Dynamically-typed computations for order-sorted equational presentations

Claus Hintermeier, Hélène Kirchner, Claude Kirchner

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

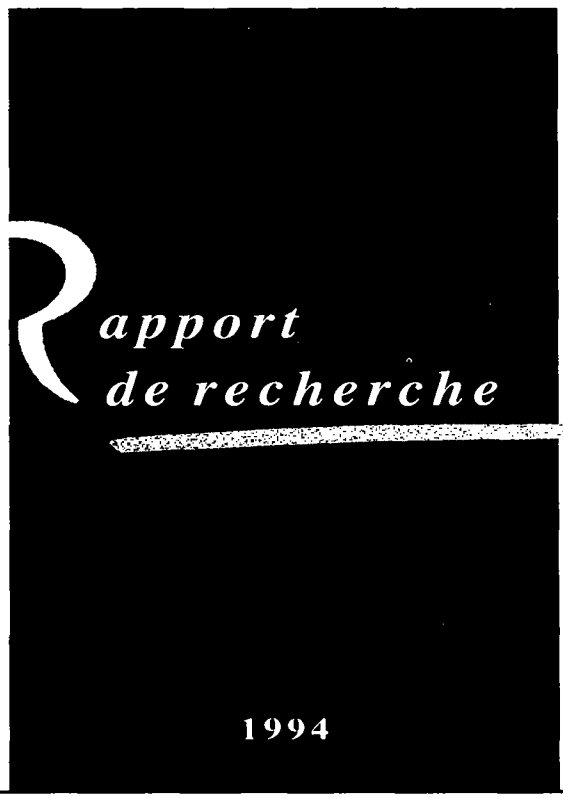
Dynamically-Typed Computations for Order-Sorted Equational Presentations

Claus HINTERMEIER
Claude KIRCHNER - Hélène KIRCHNER

N° 2208
Mars 1994

PROGRAMME 2

Calcul symbolique,
programmation
et génie logiciel



Dynamically-Typed Computations for Order-Sorted Equational Presentations

Claus Hintermeier, Claude Kirchner, Hélène Kirchner *

CRIN & INRIA-Lorraine
BP 239, 54506 Vandœuvre-lès-Nancy Cedex
France
E-mail: hinterme@loria.fr, ckirchne@loria.fr, hkirchne@loria.fr

Abstract

Equational presentations with ordered sorts encompass partially defined functions and subtyping information in an algebraic framework. In this work we address the problem of computing in order-sorted algebras, with very few restrictions on the allowed presentations. We adopt the G-algebra framework, where equational, membership and existence formulas can be expressed, and that provides a complete deduction calculus which incorporates the interaction between all these formulas.

To practically deal with this calculus, we introduce an operational semantics for G-algebra using rewrite systems over so-called decorated terms, that have assertions concerning the sort membership of any subterm in its head node. Decorated rewrite rules perform equational replacement, decoration rewrite rules enrich the decorations and record sort information. Therefore we use the semantic sort principle, i.e. equal terms belong to equal sorts, rather than the syntactic sort principle that does not use the equational part of a presentation.

In order to have a complete and decidable unification on decorated terms, we restrict to sort inheriting theories. The sort inheritance property is undecidable in general but we provide a test to check it on a given presentation. The test provides information on how to extend the presentation in a model conservative way, in order to obtain sort inheritance .

Then a completion procedure on decorated terms is designed to compute all interactions between equational and membership formulas. When the completion terminates, the resulting set of rewrite rules provides a way to decide equational theorems of the form $(t = t')$ and typing theorems of the form $(t : A)$.

*This work has been partially supported by the Esprit Basic Research working group COMPASS and by the GDR PAOIA of CNRS.

Calculs avec typage dynamique pour les théories équationnelles avec sortes ordonnées

Claus Hintermeier, Claude Kirchner, Hélène Kirchner
CRIN & INRIA-Lorraine,
BP 239, 54506 Vandœuvre-lès-Nancy Cedex,
France,
E-mail: hinterme@loria.fr, ckirchne@loria.fr, hkirchne@loria.fr

Résumé

Les présentations équationnelles avec sortes ordonnées prennent en compte, dans un cadre algébrique, les fonctions partielles et l'information de sous-types. Dans ce papier, nous considérons le problème de calculer, avec le moins de restrictions possible, dans les algèbres avec sortes ordonnées. Nous nous plaçons dans le cadre de la G-algèbre, où des formules équationelles, d'appartenance et d'existence peuvent être exprimées, et qui fournit un système de déduction complet incorporant l'interaction entre toutes ces formules.

Pour calculer en pratique avec ce système, nous introduisons une sémantique opérationnelle pour la G-algèbre, utilisant des systèmes de réécriture sur des termes décorés, qui contiennent dans chaque nœud des assertions concernant l'appartenance du sous-terme à certaines sortes. Des règles décorées réalisent le remplacement équationnel, tandis que des règles de décoration enrichissent les décorations et enregistrent les informations de sortes. Nous utilisons donc le principe des sortes sémantiques, i.e. des termes égaux appartiennent à des sortes identiques, plutôt que des sortes syntaxiques.

Afin d'obtenir une unification complète et décidable sur les termes décorés, nous nous restreignons à des théories avec héritage de sortes. Cette propriété est indécidable en général, mais nous proposons un test pour la vérifier sur une présentation donnée. Ce test fournit en cas d'échec une indication sur la façon d'étendre la présentation en conservant ses modèles, pour obtenir l'héritage de sortes.

Une procédure de complétion sur les termes décorés calcule toutes les interactions entre formules équationnelles et formules d'appartenance. Quand la complétion termine, l'ensemble de règles résultant fournit un processus de décision pour les théorèmes équationnels de la forme $(t = t')$ et pour les théorèmes de typage de la forme $(t : A)$.

Contents

1 Introduction

Due to the need of dealing with partial functions, types and polymorphism, in particular in algebraic specifications [EM85, BLK+90] but also in computer algebra [JS92, Mio93], the two last decades have seen the emergence of many frameworks integrating the notions of sort, subsort, equality and polymorphism, both in the first and higher-order contexts [CG91]. In this work, we address the problem of computing efficiently in order-sorted algebras with the less restrictive conditions on the allowed presentations.

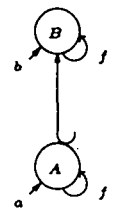
1.1 The Problems

In most studies on order-sorted computations, a logic is defined and the notion of model is introduced accordingly, together with results stating the correspondence between the model theoretic and the proof theoretic levels. Let us mention, without any exhaustivity, the work of [Obe62] on order-sorted logic, the introduction of order-sorted algebras in [Gog78], fully developped in [GM92], the order-sorted semantics introduced by [SNGM89] and the term declarations used in [SS87]. Actually, these works assume statements of the form $A \leq B$ (sort inclusion declaration) and $f : A\ B \rightarrow C$ (operator declaration), to be parsing-oriented declarations. They are used to define well-formed terms and are not directly encompassed in the deduction process. This results in various problems to get a Birkhoff-like completeness theorem or to decide local confluence [JKKM92, Wal92].

A typical situation is as follows: Suppose that A is a subsort of the sort B, the operator f is declared as operating only from A to itself, a is a constant in A and b is a constant in B. If the equality $a = b$ is given, it is problematic to deduce $f(a) = f(b)$ since, considering the previous declarations only as parsing assumptions, $f(b)$ is not a correctly constructed term. From this point of view, equality is no more a congruence. This problem has been overcome in several ways, for example in [GM92] by adding restrictions to the signatures (such as coherence and regularity), or by considering the concept of dynamic sort [WD89].

Another strange behavior of the approach considering sort inclusions and operator declarations as syntactic assertions, is that the standard rewriting tools do not behave as expected. The following specification, written below in an OBJ-like syntax and given by [SNGM89], presents a specification with a rewrite system that has no critical pair but is not locally confluent.

sort $A\ B$
subsort $A \leq B$
op $f : A \rightarrow A$
op $f : B \rightarrow B$
op $a : \rightarrow A$
op $b : \rightarrow B$
rule : $a \rightarrow b$
rule : $f(x : A) \rightarrow x$



There is no (standard) critical pair between the two rewrite rules (or oriented equalities), but the term $f(a)$ can be rewritten into a using the second rule, then to b, and into $f(b)$ using the first one. But both b and $f(b)$ are then irreducible. In this case, the rewriting process is not smart enough to discover the fact that since $a \rightarrow b$ and $a : A$ then b is also of sort A, which allows making the peak convergent via the application of $f(x : A) \rightarrow x$. A simple way to overcome the problem is to restrict to sort-decreasing rules [KKM88], that is rewrite rules that always decrease the sort of the rewritten term. The problem encountered by this first approach to order-sorted computations is that terms may be syntactically ill-formed although semantically correct. A proposed operational solution, described

in [GJM85, JKKM92], is to add some retracts at execution time but this does not solve the problem in full generality.

Completion procedures for order-sorted algebraic specifications have already been proposed. For example, in [GKK90], the order-sorted algebra framework from [GJM85, KKM88] is considered. Another framework for order-sorted computations is developed in [SNGM89]. For a comparison between the two previous approaches, see [Wal92]. In [Gan91], a translation from order-sorted to conditional many-sorted specifications is proposed as an operational solution for order-sorted completion. All these approaches need the additional hypothesis of sort-decreasing rules, in order to prove an adequate version of the critical pair lemma.

More recent works on deduction with constraints [KKR90, Com92] lead to consider this last problem as an instance of the more general phenomenon of interaction between constraints and formulas. Typically, the rules of the above example can be expressed as follows:

$$\begin{array}{ll} a & \rightarrow b \\ f(x) & \rightarrow x \ \| \ x \in A \end{array}$$

where the sort constraint $x \in A$ accounts for the typing information $(x : A)$.

Restrictions such as regularity, coherence and sort-decreasingness, can be understood as a way to solve the interaction between sort constraints and formulas. But this is not quite satisfactory, and alternative solutions to circumvent the problem sketched in the example above, have been searched. A first direction is to modify the unification and matching operations so that they become more powerful. This is the approach developed by H. Comon, using in particular second-order monadic unification [Com92]. We propose here another approach, which relies on a general framework called G-algebra introduced in [Még90] and allowing sorts, subsorts, equations and partial functions in a first-order setting. This leads to a new logic and a notion of models briefly described in Section 3. In particular the problem of the meaning of $f(a) = f(b)$ in the above example is overcome, since it is possible to infer from the presentation that b, being equal to a, is also of sort A. In such a context, one can also write for example that $i * i : \mathbf{R}$ where i is the well-known complex number, and terms like $pop(pop(push(a, push(a, P))))$ evaluate in a natural way to P, without the help of retracts. Indeed in G-algebra [Még90], typing becomes proving, and the deal is to automatize these proofs and to achieve efficient computations in this framework.

1.2 Our Approach

With the previous motivations and ideas in mind, our goals in this work are first to introduce as few restrictions on the order-sorted presentations as possible, second to deeply understand why some restrictions are needed to perform complete deduction in the proposed framework, and third to give further hints on additional requirements that could be introduced to get better efficiency of order-sorted computations.

In G-algebra, formulas are equalities $(t = t')$ or term declarations $(t : A)$ (also called membership formulas) and the deduction process incorporates interaction between these two kinds of formulas. But this generality leads to undecidability of typing, precisely because of the interaction between sort and equality computations, and since non trivial term declarations are allowed. This has two immediate consequences: matching and unification are undecidable in G-algebra. Thus, it is impossible to apply the usual technique for simulating equational logic with rewriting logic [Mes92] via term rewriting completion, since rewriting (using matching) as well as superposition of rules (using unification) is undecidable.

To cope with this undecidability problem, we propose an operational semantics for the deduction rules in G-algebra given in [Még90], based on an adequate representation of terms, called decorated terms, and on a translation of the presentation formulas into rewrite rules.

The dynamic aspect of sort information and its interaction with equality is taken into account through the notion of decorated terms. Decorations are sets of sorts, recording the currently proved sorts of a term. They are spread out in the term and act locally as sort constraints during the deduction. Managing sort information locally at each node of a term imposes the definition of adequate matching and unification. Decidable matching on decorated terms allows defining rewriting with two appropriate notions of rewrite rules. The first kind of rules corresponds to the equality axioms of the presentation, turned into *decorated rules* that rewrite terms and possibly enrich their decorations. The second one corresponds to term declarations turned into conditional *decoration rules* that only enrich decorations without modifying the term structure.

Using these tools, we then come back to the semi-decision problem of equality formulas $(t = t')$ and membership formulas $(t : A)$, in G-algebra. This is achieved via a generalization of the well-known completion procedure [KB70] which works on decorated terms using the decidable versions of matching and unification mentioned before. Doing so of course, we push all the underlying undecidability problems at the completion level.

The completion process that we propose is based on the hypothesis that the information in the presentation is modularized in three parts. The first one (i) consists in all equalities $(t = t')$, the second one (ii) consists in all term declarations $(t : A)$ and the last one (iii) in the sort ordering structure defined by sort inclusions $(A \leq B)$. The two first ones are handled via rewriting rules and are thus modified and enriched during completion. On the contrary, the third one is considered stable during the *whole* completion. In particular, matchers and unifiers are computed using as usual the term structure but also the sort information given in decorations and in the –fixed– sort structure. Since matching and unification use only the sort information available in the decorated term at unification or matching time, they are correct but non complete in general. Therefore, in order to get completeness for the critical pairs computation involved into completion, it is necessary that the sort information given in part (iii) contains enough information to have the following property: if a term t can be proven in the presentation to be of sorts A and B, then these two sorts have a non-empty common subsort. This completeness property of the sort information part (iii) of the presentation is called *sort inheritance* and we assume it true all over the completion.

The completion process is performed assuming the sort inheritance of the presentation, necessary for the completeness of the critical pairs computation. If completion terminates, the resulting set of rewrite rules provides a way to prove not only equational theorems of the form $(t = t')$ but also typing theorems of the form $(t : A)$.

We then provide a procedure to check sort inheritance of a presentation. When sort-inheritance is not satisfied, this is detected by a failure of this process. A counter-example is then provided that can be exploited to enrich the sort structure.

It is worth emphasizing that in this approach, typing information has two parts: the static part contains only the subsort relation. The dynamic part covers the term declarations in form of rewrite rules, which are handled outside of unification and provide automatically a typing algorithm in the completed presentation.

As a consequence of this approach, we get quite general order-sorted computations that can be very efficiently implemented because of the memorization of sort information in the terms and the static behavior of matching and unification. Moreover the notion of retract is not needed anymore to deal with syntactically ill-formed terms.

1.3 A Simple Example

Applying this approach to the previous example leads to the following steps of computations. The presentation is first translated into G-algebra formulas (second column below):

$sort\ A\ B$	$x :: A,\ y :: B$
$subsort\ A \leq B$	$x : B$
$op\ f : A \to A$	$f(x) : A$
$op\ f : B \to B$	$f(y) : B$
$op\ a : \to A$	$a : A$
$op\ b : \to B$	$b : B$
$rule : a \to b$	$a = b$
$rule : \forall x : A,\ f(x) \to x$	$f(x) = x.$

Then, using the following set of decoration rules whose construction is explained later on, and where the variable s can be instantiated by any set of sorts, the formulas of the first column are translated in the formulas of the second one:

$f(x) : A$	$f(x^{:\{A\}})^{:s}$	$\to$	$f(x^{:\{A\}})^{:s\cup\{A\}}$ if $\{A\} \not\subseteq s$
$f(y) : B$	$f(y^{:\{B\}})^{:s}$	$\to$	$f(y^{:\{B\}})^{:s\cup\{B\}}$ if $\{B\} \not\subseteq s$
$a : A$	$a^{:s}$	$\to$	$a^{:s\cup\{A\}}$ if $\{A\} \not\subseteq s$
$b : B$	$b^{:s}$	$\to$	$b^{:s\cup\{B\}}$ if $\{B\} \not\subseteq s.$
$a = b$	$a^{:\emptyset}$	$=$	$b^{:\emptyset}$
$f(x) = x$	$f(x^{:\{A\}})^{:\emptyset}$	$=$	$x^{:\{A\}}$

The presentation is saturated, using a completion process, into the following one:

$$
\begin{aligned}
f(x^{:\{A,B\}})^{:s} &\to f(x^{:\{A,B\}})^{:s\cup\{A,B\}} \text{ if } \{A,B\} \not\subseteq s \\
f(y^{:\{B\}})^{:s} &\to f(y^{:\{B\}})^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
a^{:s} &\to a^{:s\cup\{A,B\}} \text{ if } \{A,B\} \not\subseteq s \\
b^{:s} &\to b^{:s\cup\{A,B\}} \text{ if } \{A,B\} \not\subseteq s \\
a^{:\{A,B\}} &\to b^{:\{A,B\}} \\
f(x^{:\{A,B\}})^{:\{A,B\}} &\to x^{:\{A,B\}}
\end{aligned}
$$

In this saturated presentation, the term $f(a^{:\emptyset})^{:\emptyset}$ is first decorated (using the decoration rules) into $f(a^{:\{A,B\}})^{:\{A,B\}}$. Then it is rewritten using the two last decorated rewrite rules above into $b^{:\{A,B\}}$.

Note that in this new framework the restrictions of regularity, coherence and sort-decreasingness are not needed any more to get the usually expected results.

1.4 Structure of the Paper

The paper is organized in the following way. First the G-algebra framework is recalled. Then the definition of decorated terms is introduced, with the corresponding matching and unification notions. Based on this matching, we define rewriting with decorated rewrite rules and with decoration rewrite rules. A completeness theorem states the equivalence of replacement of equal by equal on decorated terms with deduction in G-algebra. Using unification on decorated terms, critical pairs are defined between the two different kinds of rules. A completion process that involves both kinds of rules is then described and we show how the completed presentation allows proving equational or typing theorems, when the initial presentation is sort inheriting. A process to check sort inheritance is then proposed. Our approach is eventually compared with other related works.

2 Basic Notions and Notations

Notations concerning classical terms, occurrences, replacements, substitutions and generality on terms and substitutions are consistent with [DJ90, SNGM89]. In particular we write $|t|$ for the number of function symbols occurring in a term t, $subterm\_set(t)$ for all subterms of t, $\mathcal{VO}cc(t)$ for the set of all variable occurrences, $\mathcal{VO}cc_x(t)$ for the set of all occurrences of the variable x and $\mathcal{NVO}cc(t)$ for the non-variable ones in t.

We refer to [DJ90] in particular for the definition of the various orderings used in this paper. Specifically, lexicographic orderings are written as a tuple $(\leq_1, \ldots, \leq_n)$ of the orderings $\leq_1, \ldots, \leq_n$.

The notion of replacement will also be used in an extended way for occurrence sets and sequential multiple replacements. Therefore, $t[u]_O$ stands for t with u at all occurrences $\omega \in O$ and $t[u]_O[v]_{O'}$ for $(t[u]_O)[v]_{O'}$. Furthermore, the occurrence orderings extend pairwise to occurrence sets, i.e. O is incomparable with O' iff all occurrences in O are incomparable with all occurrences in O' etc. Additionally, the concatenation of two occurrence sets O and O' denotes the set of occurrences $\{\omega.\omega' \mid \omega \in O \text{ and } \omega' \in O'\}$. Finally, if $t_{|\omega_1} = t_{|\omega'}$ for all $\omega, \omega' \in O$, then $t_{|O}$ stands for $t_{|\omega}, \omega \in O$.

For a substitution σ, $\mathcal{D}om(\sigma) = \{x \mid \sigma(x) \neq x\}$, $\mathcal{R}an(\sigma) = \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(\sigma(x))$ is the set of variables in the image of the variables in the domain of the substitution, and $\mathcal{I}m(\sigma)$ stands for the set $\{t \mid \exists x \in \mathcal{D}om(\sigma) : \sigma(x) = t\}$. Moreover, $terms(\rho)$ stands for the terms occurring in an arbitrary structure ρ without its subterms and $subterm\_set(\rho)$ is $terms(\rho)$ together with all subterms of terms in $terms(\rho)$.

All notions concerning unsorted matching, unification and term rewriting are consistent with those in [JK91]. Notions on completion, equational proofs and proof reductions are consistent with [Bac91].

3 Short Introduction to G-Algebra

In this section we briefly define formulas and presentations in G-algebra. A main feature of this framework is that term declarations, usually seen as part of the (static) signature in classical approaches, become G-algebra formulas and are involved in proofs at the same level as equalities.

3.1 Formulas and Presentations

Let $\mathcal{S}$ be a set of sort symbols, containing always the universal sort symbol Ω. For $\mathcal{X}$ a set of variables, '::' is a binary relation associating to any variable a unique sort in $\mathcal{S}$, denoted $sort(x)$. Then $\mathcal{X}$ is called a $\mathcal{S}$-sorted set of variables. Let also $\mathcal{F}$ be a set of function symbols with an arity function $arity()$ defined for each element of $\mathcal{F}$. Then $\Sigma = (\mathcal{S}, \mathcal{F})$ is called *signature*. A $(\Sigma, \mathcal{X})$-*term* is either a variable $x \in \mathcal{X}$, or of the form $f(t_1, \ldots, t_n)$ with $f \in \mathcal{F}$, $ar(f) = n$ and $t_1, \ldots, t_n$ being $(\Sigma, \mathcal{X})$-terms. The set of all $(\Sigma, \mathcal{X})$-terms is denoted by $\mathcal{T}(\Sigma, \mathcal{X})$, the ground $(\Sigma, \mathcal{X})$-terms by $\mathcal{T}(\Sigma)$. Notice that except the condition on arity, there is no requirement of well-sortedness: the fact that a term is well-sorted will result from an (arbitrary complicated) proof and is thus a semantical fact rather than a syntactical one.

A formula in a G-algebra can be an existence formula for a term t, written $(EX\ t)$, a membership formula for t to be in a sort A, written $(t : A)$, a variable declaration for x to be of sort A, written $(x :: A)$, or an equality of two terms t and t', written $(t = t')$. All formulas are implicitly universally quantified over all variables occurring in the formula. In the G-algebra framework, a set of formulas built on a signature Σ is called Σ-*presentation*. A pair $(\Sigma, \mathcal{P})$ of a signature Σ and a Σ-presentation is called *specification*.

For every signature Σ, a Σ-algebra $\mathcal{A}$ is defined by its domain $|\mathcal{A}|$ and by interpretations for each symbol in $\mathcal{S}$ and $\mathcal{F}$:

1. $\forall A \in \mathcal{S}$, the interpretation of A, $A^{\mathcal{A}}$ is a non-empty set, and $\Omega^{\mathcal{A}} = |\mathcal{A}|$,

2. $\forall f \in \mathcal{F}$, the interpretation of f, $f^{\mathcal{A}}$ is a partial function $f^{\mathcal{A}} : |\mathcal{A}|^{arity(f)} \rightarrow |\mathcal{A}|$.

A *variable assignment* α of variables in a set $\mathcal{V} \subseteq \mathcal{X}$ in a Σ-algebra $\mathcal{A}$ is a function that assigns for all $x \in \mathcal{V}$, $(x :: A) \in \mathcal{P}$, an element in $A^{\mathcal{A}}$. Hence, given a Σ-algebra, where $\equiv$ is the equality relation, formulas can be interpreted in the following way:

1. $\mathcal{A} \models (EX\ t)$ if there is an assignment for all variables in t and for every assignment α of the variables in t, $\alpha(t) \in \Omega^{\mathcal{A}}$
2. $\mathcal{A} \models (t : A)$ if there is an assignment for all variables in t and for every assignment α of the variables in t, $\alpha(t) \in A^{\mathcal{A}}$
3. $\mathcal{A} \models (t = t')$ if there is an assignment for all variables in t and t' and for every assignment α of the variables in t and t', $\alpha(t) \equiv \alpha(t')$

A Σ-algebra $\mathcal{A}$ is a *model* of a Σ-presentation $\mathcal{P}$ if for each ϕ in $\mathcal{P}$, $\mathcal{A} \models \phi$. $\mathcal{P} \models \phi$ if ϕ is true in all models of $\mathcal{P}$. Note that if a function is not defined for some argument, the result of this function application is not in the interpretation of any sort and two undefined results of function applications are never equal. A Σ-term algebra for a Σ-presentation is a Σ-algebra $\mathcal{A}$ that uses the identity function as interpretation for each symbol in $\mathcal{F}$. Given a Σ-presentation $\mathcal{P}$, the reflexive, symmetric, transitive, substitutive and congruence closure of the equations in $\mathcal{P}$ defines a congruence $\equiv$. The quotient term algebra $\mathcal{T}_{\Sigma/\equiv}$, is a Σ-algebra which is proved initial in the class of Σ-algebras satisfying the presentation $\mathcal{P}$ [Még90].

3.2 Deduction

The deduction rules for G-algebras are shown in Figure 1. When a formula ϕ can be deduced from the formulas of a presentation $\mathcal{P}$ using these deduction rules, this is denoted $\mathcal{P} \vdash \phi$. The substitutions σ mentioned in the deduction system are supposed to be *conform* with the current presentation which means that, for all $(x \mapsto t) \in \sigma$, we have $\mathcal{P} \vdash (t : A)$ if $(x :: A) \in \mathcal{P}$. These rules are proved to be sound and complete in [Még90] i.e. for any presentation $\mathcal{P}$ and formula ϕ:

$$\mathcal{P} \models \phi \Leftrightarrow \mathcal{P} \vdash \phi.$$

The number of rules for the complete and sound deduction system has increased with respect to the unsorted or many-sorted cases, because membership proofs have the same status as equality and existence proofs. Notice also that membership to Ω and existence formulas are equivalent statements and thus this set of rules can be simplified if needed.

4 Sort Membership

4.1 Associating a Partial Ordering over Sorts

In the following sections, we use a quasi-ordering $\leq_S^{syn}$ over the sorts, which is extracted from the variable declarations in the current presentation $\mathcal{P}$. This simplifies notations but mainly allows for a modularization of deduction. In particular the unification process will heavily rely on this quasi-ordering.

Definition 4.1 *Let $\mathcal{P}$ be a presentation. The* syntactic sort ordering in $\mathcal{P}$, *written $\leq_S^{syn}$, is the transitive and reflexive closure of the relation:*

Globality	$EX\ t$	$\Longrightarrow$	$t : \Omega$
VariableMembership	$x :: A$	$\Longrightarrow$	$x : A$
ExSubterm	$EX\ t[u]$	$\Longrightarrow$	$EX\ u$
ExMembership	$t : A$	$\Longrightarrow$	$EX\ t$
ExEquality	$t = t'$	$\Longrightarrow$	$EX\ t$
ExReplacement	$EX\ t[u], u = v$	$\Longrightarrow$	$EX\ t[v]$
MeReplacement	$t[u] : A, u = v$	$\Longrightarrow$	$t[v] : A$
EqReplacement	$t[u] = w, u = v$	$\Longrightarrow$	$t[v] = w$
ExSubstitutivity	$EX\ t$	$\Longrightarrow$	$EX\ \sigma(t)$
MeSubstitutivity	$t : A$	$\Longrightarrow$	$\sigma(t) : A$
EqSubstitutivity	$t = t'$	$\Longrightarrow$	$\sigma(t) = \sigma(t')$
Reflexivity	$EX\ t$	$\Longrightarrow$	$t = t$
Symmetry	$u = v$	$\Longrightarrow$	$v = u$
Transitivity	$u = v, v = w$	$\Longrightarrow$	$u = w$

DeduceGAlgebra

Figure 1: Deduction rules for G-algebra

$$A \leq_{\mathcal{S}}^{syn} B \text{ if } \exists \{x :: A,\ x : B\} \subseteq \mathcal{P}.$$

A is called subsort *of B, if* $A \leq_{\mathcal{S}}^{syn} B$. *If* $A \leq_{\mathcal{S}}^{syn} B$ *and* $B \leq_{\mathcal{S}}^{syn} A$, *we write* $A \sim_{\mathcal{S}}^{syn} B$. *The negation is written* $A \not\sim_{\mathcal{S}}^{syn} B$. *When* $A \leq_{\mathcal{S}}^{syn} B$ *but* $A \not\sim_{\mathcal{S}}^{syn} B$, *A is a* strict subsort *of B, written* $A <_{\mathcal{S}}^{syn} B$. *If neither* $A \leq_{\mathcal{S}}^{syn} B$ *nor* $B \leq_{\mathcal{S}}^{syn} A$, *A and B are said* incomparable, *written* $A \bowtie_{\mathcal{S}}^{syn} B$.

The upward closure *of a sort A, written* $A\uparrow$, *is the set* $\{B \mid A \leq_{\mathcal{S}}^{syn} B\}$.

The (semantic) sort ordering in $\mathcal{P}$, *written* $\leq_{\mathcal{S}}^{sem}$, *is the transitive and reflexive closure of the following relation:*

$$A \leq_{\mathcal{S}}^{sem} B \text{ if } \exists (x :: A) \in \mathcal{P} \text{ such that } \mathcal{P} \models x : B,$$

Clearly, $\leq_{\mathcal{S}}^{syn} \subseteq \leq_{\mathcal{S}}^{sem}$, but not always $\leq_{\mathcal{S}}^{syn} = \leq_{\mathcal{S}}^{sem}$, as the following example illustrates:

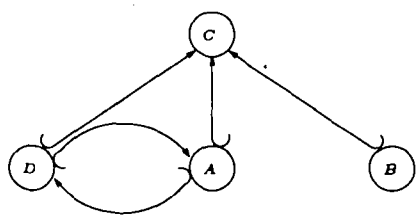
Example 4.2 *Let* $\mathcal{P} = \{x :: A, y :: B, f(x,y) : B, f(x,y) = x\}$. *Clearly,*

$$\mathcal{P} \models x : B \text{ and } x :: A \in \mathcal{P}.$$

Therefore $A \leq_{\mathcal{S}}^{sem} B$, *but* $A \leq_{\mathcal{S}}^{syn} B$ *is not true.*

The following example illustrates the notions that we just introduced:

Example 4.3 *Let* $\mathcal{P} = \{x :: A, x : C, x : D, y :: B, y : C, z :: C, u :: D, u : A, x = y\}$.
Then $A <_{\mathcal{S}}^{syn} C$, $B <_{\mathcal{S}}^{syn} C$, $D \sim_{\mathcal{S}}^{syn} A$, $D <_{\mathcal{S}}^{syn} C$, $A \bowtie_{\mathcal{S}}^{syn} B$ *and* $D \bowtie_{\mathcal{S}}^{syn} B$. *This is represented by the following graph:*



Therefore, we also have $A <^{sem}_{\mathcal{S}} C$, $B <^{sem}_{\mathcal{S}} C$, $D \sim^{sem}_{\mathcal{S}} A$, $D <^{sem}_{\mathcal{S}} C$, $A \sim^{sem}_{\mathcal{S}} B$ *and* $D \bowtie^{sem}_{\mathcal{S}} B$. *Note that* $\mathcal{P} \models x : B, y : A$, *but neither* $A \leq^{syn}_{\mathcal{S}} B$ *nor* $B \leq^{syn}_{\mathcal{S}} A$.

Unfortunately, as one can expect, we get the following negative result for the semantic relation $\leq^{sem}_{\mathcal{S}}$:

Proposition 4.4 *It is undecidable whether for an arbitrary given specification* $((\mathcal{S}, \mathcal{F}), \mathcal{P})$, *and two sorts* $A, B \in \mathcal{S}$, $A \leq^{sem}_{\mathcal{S}} B$ *holds.*

Proof: We actually show that $A \sim^{sem}_{\mathcal{S}} B$ is undecidable. This is sufficient since assuming $\leq^{sem}_{\mathcal{S}}$ decidable implies immediately that $\sim^{sem}_{\mathcal{S}}$ is decidable, too.

It is well-known, that there exist undecidable equational proofs, e.g. in equational theories containing the Turing machine semantics (see [Dau89]). Let $((\mathcal{S}, \mathcal{F}), \mathcal{P})$ be a specification of such a theory and $(t = t')$ an undecidable equality. Let $\delta\mathcal{S} = \{A, B\}$, where A, B are not in $\mathcal{S}$ and therefore not used in $\mathcal{P}$, and $\delta\mathcal{P} = \{t = x, x :: A, t' = x', x' :: B\}$. Thus $A \sim^{sem}_{\mathcal{S}} B$ in $((\mathcal{S} \cup \delta\mathcal{S}, \mathcal{F}), \mathcal{P} \cup \delta\mathcal{P})$ is undecidable, since it is equivalent to deciding $(t = t')$ in $((\mathcal{S}, \mathcal{F}), \mathcal{P})$. □

4.2 Assumptions Concerning the Sort Membership

In order to keep the unification of two variables decidable, we restrict the used signatures to sort inheriting ones. This is a more semantical notion than classical regularity, i.e. the existence of a unique least sort for each term. Sort inheritance just means that if a term can be proved (semantically) to be of sorts A and B, then these two sorts have a common subsort:

Definition 4.5 *Let* $\Sigma = (\mathcal{S}, \mathcal{F})$ *be a signature. A specification* $(\Sigma, \mathcal{P})$ *is* sort inheriting *if* $\forall t \in \mathcal{T}(\Sigma, \mathcal{X}) : \forall A, B \in \mathcal{S}$:

$$\mathcal{P} \vdash t : A, t : B \Rightarrow \exists C \in \mathcal{S} : C \leq^{syn}_{\mathcal{S}} A, B.$$

Another way this definition can be understood is that a specification $(\Sigma, \mathcal{P})$ is sort inheriting if the sort information present in $\mathcal{P}$ is complete w.r.t. the semantic sort membership.

Altogether, we make the following general assumptions in all the paper.

General Assumption 4.6

4.6.1 *The sort relation does not contain cycles.*

4.6.2 *The set* $mlb(S)$ *of maximal elements of the set of lower bounds of any subset of sorts* $S \subseteq \mathcal{S}$ *is computable.*

4.6.3 *All sorts are non-empty.*

4.6.4 *The presentation is sort inheriting.*

4.6.5 *The specification has bounded membership, i.e.* $\#\{A \mid \mathcal{P} \vdash t : A\}$ *is finite for all* $t \in \mathcal{T}(\Sigma, \mathcal{X})$.

In signatures with finite sort sets, the points 4.6.1, 4.6.2 and 4.6.5 are obviously decidable [SS87]. However, if polymorphic signatures are used, more sophisticated properties and sort concepts have to be introduced (see [Smo89]). Point 4.6.3 is undecidable in general but can be enforced by the stricter but decidable requirement that all sorts are syntactically non-empty. Finally, point 4.6.4 is undecidable in general but a constructive test exhibiting terms that destroy sort inheritance is developed in this paper. Of course all specifications whose set of sorts is finite have the bounded membership property. For polymorphic sort structures, sufficient conditions for this point have to be developed.

We give two non-regular examples to explain the difference between regularity and sort inheritance in polymorphic signatures.

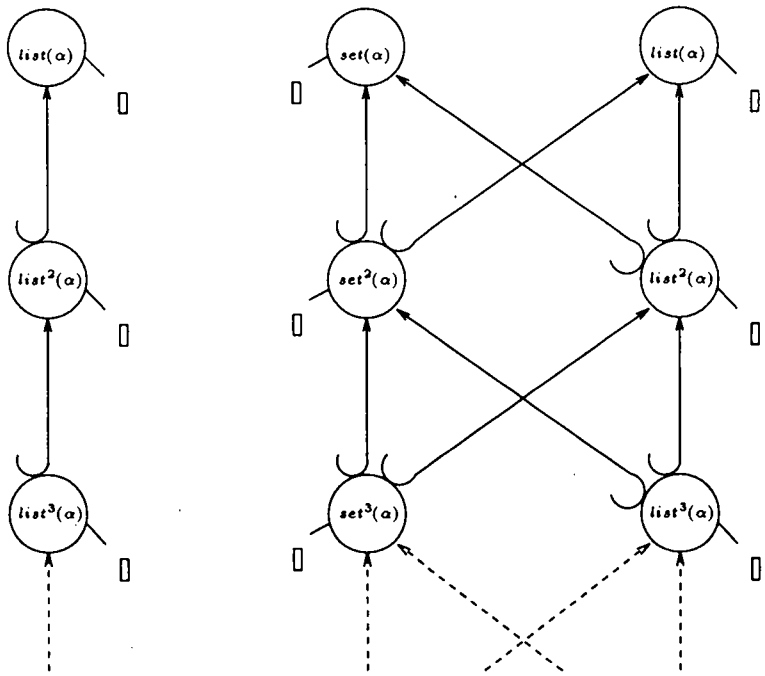


Figure 2: Two non-regular but sort inheriting sort structures

Example 4.7 *Let list(α) denote the sort constructor for the lists over arbitrary objects of sort α and* [] *the empty list, occurring in each of the list(α)-sorts. So the first structure illustrated in figure 2 is sort inheriting but not regular.*

Example 4.8 *The second structure in figure 2 extends the first example by adding a sort constructor for α-term sets called set(α), that is realized in this special case as a list with two elements and thus is a subsort of list(α). So we would like to represent the empty set with the empty list and thus we know that* [] *is contained in each of the two sorts list(α) and set(α), giving a more complex structure.*

In each of the two cases the precondition of classical unification algorithms, namely regularity, is not fulfilled, but unification is clearly decidable. Another solution to the problem of non-regularity in these cases would be the introduction of a new sort *nil* being subsort of all others and containing only [], i.e. the signature would become regular. This technique could even be extended to all elements of a polymorphic sort being independent of the sort's parameters, that are consequently elements of all instances of the polymorphic sort. However, our approach seems to be closer to the intentions of the specifier. Furthermore, the sort inheritance condition is intuitively easy to understand, even though the formal definition seems to be complicated. In general these infinite sort structures[1] also resemble a lot at the structures used in unified algebras (see [Mos89] for further details), although there is no difference between the notions of sort and term. Comparing unified and G-algebras in more detail will surely be one of the next steps concerning this approach.

Another problem in polymorphic sort theories, is the number of sorts, a term can belong to. Therefore, we have to characterize suitable specifications.

[1] They are not necessarily lattices, as the second example shows.

Definition 4.9 *A specification* $((\mathcal{S},\mathcal{F}),\mathcal{P})$ *has* bounded membership *iff* $\#\{A \mid \mathcal{P} \vdash t : A\}$ *is finite for all* $t \in \mathcal{T}(\Sigma,\mathcal{X})$.

Of course all specifications whose set of sort is finite have the bounded membership property.

For polymorphic sorts theories, sufficient conditions are still to be introduced. A weakening could be reached by demanding a finite representation of all sorts of a term.

4.3 Eliminating cycles

For the resolution of peaks by critical pairs, presentations without cycles in the sort structure will be required. In the case of finite sort sets the calculation of minimal, complete subsort sets of two sorts is unique, i.e. the unification algorithm is also more efficient.

The transformation of presentations replaces all sort symbols occurring in such a cycle by a unique representative of the sort symbols in the cycle. The theory described by the presentation remains the same, because G-algebras force the models of a cyclic presentation $\mathcal{P}$ by the rule **MeSubstitutivity** to include the domain of a sort S_i in the domain of a sort S_{i+1}, if $\{x_i :: S_i, x_i : S_{i+1}\} \subseteq \mathcal{P}$. Hence, all the inclusions of sort domains in a cycle cause the equality of the sort domains for all sorts occurring in the cycle.

We start with the formalization of cycles in the variable declaration part of a presentation $\mathcal{P}$, denoted in the following by $\mathcal{P}_V$.

Definition 4.10 *A presentation is called* cyclic in the variable declaration part *or just* cyclic, *if there is a finite chain of declarations* $\{x_i : S_i, x_i :: S_{i+1}\}_{i\in[1..n]}$ *in* $\mathcal{P}_V$ *with* $S_1 = S_n$. $C \stackrel{def}{=} \bigcup_{i\in[1..n]}\{S_i\}$ *is called* cycle *and a sort* S *is* member of a cycle, *written* $S \in C$, *if there is a* $i \in [1..n]$, *s.t.* $S = S_i$.

The cycles of a finite presentation can easily be found with one of the various algorithms existing in graph theory. The easiest one is surely the bounded depth first search with $|\mathcal{S}|$ as limit. For the real transformation we wish to find a minimal set containing all cycles included in the variable declaration part of the current presentation.

Definition 4.11 *Let* $\mathcal{C}$ *be the set of all cycles contained in* $\mathcal{P}_V$. *Then the set* $\mathcal{C}_c$ *is called a* minimal, complete set of cycles in $\mathcal{P}_V$, *if:*

- $\forall C \in \mathcal{C}\ \exists C_c \in \mathcal{C}_c\ :\ C \subseteq C_c$ *(completeness)*
- $\mathcal{C}_c \subseteq \mathcal{C}$ *(soundness)*
- $\#\mathcal{C}_c$ *is minimal* *(minimality)*

Note that such a set of cycles does not necessarily exist. Just think of an infinite set of sorts and presentations with an infinite number of disjoint cycles. But if such a set exists, it is trivially unique:

Proposition 4.12 *Let* $\mathcal{P}$ *be a presentation and* $\mathcal{C}_c$ *a minimal, complete set of cycles in* $\mathcal{P}_V$. *Then* $\mathcal{C}_c$ *is unique.*

Proof: Let $\mathcal{C}_c^1$ and $\mathcal{C}_c^2$ be two different minimal, complete sets of cycles in $\mathcal{P}_V$ and let c be in $\mathcal{C}_c^1$ but not in $\mathcal{C}_c^2$. $\mathcal{C}_c^2$ is complete and thus there is a $c_2 \in \mathcal{C}_c^2$, such that $c \subseteq c_2$. But $\mathcal{C}_c^1$ is also complete and therefore we have a $c_1 \in \mathcal{C}_c^1$ with $c_2 \subseteq c_1$. Furthermore, $\mathcal{C}_c^1$ is minimal, i.e. $c = c_2 = c_1$ and ergo $\mathcal{C}_c^1 = \mathcal{C}_c^2$.

□

The uniqueness of the minimal complete set of cycles in $\mathcal{P}_V$ allows us to replace all sorts in a cycle by a unique representative.

Proposition 4.13 *Let $\mathcal{C}_c = \{C_1, \ldots, C_n\}$ be the finite, minimal, complete set of cycles in $\mathcal{P}_V$ and $\mathcal{S} = \{S_1, \ldots, S_m\}$. Then the set $\mathcal{S}_c(\mathcal{C}_c)$ with:*

$$\mathcal{S}_c(\mathcal{C}_c) \stackrel{def}{=} \{S_{\Pi(i)} \mid 1 \leq i \leq m \text{ and } S_i \in C_{j(i)} \text{ and } \Pi(i) = max_{S_k \in C_{j(i)}}(k)\}$$

is isomorphic with $\mathcal{C}_c$.

Proof: The transformation of $\mathcal{C}_c$ into $\mathcal{S}_c(\mathcal{C}_c)$ is clearly the function $\mathcal{S}_c$ itself. For the inverse transformation, we can easily reconstruct the C_i from $\mathcal{C}_c$ by searching the maximal cycles in $\mathcal{P}$ starting with $S_{\Pi(i)}$. □

We can now introduce the transformation that eliminates the cycles in $\mathcal{P}_V$.

Proposition 4.14 *Let $\mathcal{P}$ be a presentation, $\mathcal{C}_c = \{C_1, \ldots C_n\}$ the finite, minimal, complete set of cycles in $\mathcal{P}_V$ and Θ the following transformation:*

$$\begin{array}{rcll}
\Theta(X) & \Rightarrow & \Theta_2(\Theta_1(X)) & \\
 & & & \\
\Theta_1(\emptyset) & \Rightarrow & \emptyset & \\
\Theta_1(\mathcal{P}' \cup \{x :: S\}) & \Rightarrow & \Theta_1(\mathcal{P}') \cup \{x :: S_{\Pi(i)}\} & \text{if } S \in C_i \\
\Theta_1(\mathcal{P}' \cup \{t : S\}) & \Rightarrow & \Theta_1(\mathcal{P}') \cup \{t : S_{\Pi(i)}\} & \text{if } S \in C_i \\
\Theta_1(\mathcal{P}' \cup \{t : S\}) & \Rightarrow & \Theta_1(\mathcal{P}') \cup \{t : S\} & \text{if } \nexists i \in [1..n] \ : \ S \in C_i \\
\Theta_1(\mathcal{P}' \cup \{X\}) & \Rightarrow & \Theta_1(\mathcal{P}') \cup \{X\} & \text{otherwise} \\
 & & & \\
\Theta_2(\mathcal{P}' \cup \{x :: S, x : S\}) & \Rightarrow & \Theta_2(\mathcal{P}' \cup \{x :: S\}) & \\
\Theta_2(X) & \Rightarrow & \Theta_2(X) & \text{otherwise}
\end{array}$$

Then $\mathcal{P} \models t : S_m$ iff $\Theta(\mathcal{P}) \models t : S_{\Pi(m)}$.

Proof: First of all we can state that for all variables x with $(x :: S) \in \mathcal{P}$ and $S \in C_i$, s.t. Θ changes the sort S, we have $\mathcal{P} \models x : S_{\Pi(i)}$ using **MeSubstitutivity** and the definition of a cycle. The same argument is trivially valid in the case of $(x :: S_{\Pi(i)}) \in \mathcal{P}$, where we can use **VariableMembership**. Therefore $\mathcal{P} \models x : S_{\Pi(i)}$ for any $(x :: S) \in \mathcal{P}$ and any $S \in C_i$. This proof is denoted by Φ_i. As S_i and $S_{\Pi(i)}$ are in the same cycle, we can also proof the inverse, i.e. for any x with $x :: S_{\Pi(i)} \in \mathcal{P}$, we can prove $x : S_i$ in $\mathcal{P}$. Let Φ_i' denote this proof in the sequel. Let furthermore $\Theta(t' : S')$ be defined as $\Theta(\{t' : S'\})$. The following is a sketch of the proof.

- $\Rightarrow$: This is trivial, since any proof Φ using $\mathcal{P}$ only has to be transformed replacing any membership formula $t' : S'$ by $\Theta(t' : S')$ without loosing its soundness.
- $\Leftarrow$: Since all membership declarations $t' : S_{\Pi(k)}$ in $\Theta(\mathcal{P})$ are images of some $t' : S_k$ in $\mathcal{P}$ under Θ, we can construct a proof Υ of $\tau(t') : S_{\Pi(k)}$ in $\mathcal{P}$, where τ replaces all $x_i \in Var(t')$ of sort S_i by variables y_i of sort $S_{\Pi(i)}$,

$$\begin{array}{l}
\Upsilon \ : \ x :: S_{\Pi(k)} \Leftrightarrow_{VariableMembership} x : S_{\Pi(k)} \Leftrightarrow^{\sigma_k}_{MeSubstitutivity} x_k : S_{\Pi(k)} \\
\qquad \Leftrightarrow^{\sigma}_{MeSubstitutivity} t' : S_{\Pi(k)} \Leftrightarrow^{\tau}_{MeSubstitutivity} \tau(t') : S_{\Pi(k)}
\end{array}$$

where $\sigma_k = \{x \mapsto x_k\}$, valid by Φ_k, σ is $\{x_k \mapsto t_k\}$, trivially valid by $t_k \in \mathcal{P}$, and finally $\tau = \{x_i \mapsto y_i\}$, valid by the Φ_is.

The resulting membership formula in $Th(\mathcal{P})$ is equivalent to the one in $Th(\Theta(\mathcal{P}))$, since we consider all formulas as implicitly universally quantified and therefore any use of formulas $x :: S_{\Pi(j)}$ of a proof in $\Theta(\mathcal{P})$ is equivalent to one using a variable x' with $x' :: S_{\Pi(j)}$ in $\mathcal{P}$. Therefore the rest of the proof in $Th(\Theta(\mathcal{P}))$ can be kept without any changes, giving a proof Ψ of $t : S_{\Pi(m)}$ in $\mathcal{P}$.

Finally, we construct the proof Φ of $t : S_m$ in $\mathcal{P}$ from Υ, using variables y of sort S_m and z of $S_{\Pi(m)}$ in $\mathcal{P}$:

$$y :: S_m \Leftrightarrow_{VariableMembership} y : S_m \Leftrightarrow^{\sigma_m}_{MeSubstitutivity} z : S_m \Leftrightarrow^{\sigma}_{MeSubstitutivity} t : S_m$$

where σ_k is $\{y \mapsto z\}$, valid by Φ'_m, and $\sigma = \{z \mapsto t\}$, valid by Ψ.

In the case of a derivation based on an application of the **Globality**-rule, the proof is trivial, since the previously performed membership proofs can always be followed by an arbitrary number of **Globality**-rule applications.

□

Corollary 4.15 *Let $\mathcal{P}$ be a presentation, $C_c = \{C_1, \ldots C_n\}$ the minimal complete set of cycles in $\mathcal{P}_V$, Θ the transformation defined in the precedent proposition and Θ_F the corresponding transformation for the formulas in G-algebras. Then:*

$$\mathcal{P} \models \Psi \text{ iff } \Theta(\mathcal{P}) \models \Theta_F(\Psi).$$

Thus we proved that Θ is a conservative transformation in the sense of [SS87] - in fact the factorization of equivalent sorts was already stated to be conservative there, but the proof sketch there was even more fuzzy and the transformation less operational than this one. Finally, we give an example for the defined transformation:

Example 4.16 *Let $\mathcal{S} = \{S_1, S_2, S_3\}$ and $\mathcal{P} = \{x :: S_1, y :: S_2, z :: S_3, x : S_2, y : S_3, z : S_1, t : S_3\}$. Then $\Theta(\mathcal{P}) = \{x :: S_3, y :: S_3, z :: S_3, t : S_3\}$ and:*

$$\mathcal{P} \models t : S_1 \text{ iff } \Theta(\mathcal{P}) \models t : S_{\Pi(1)},$$

since $\Pi(1) = 3$ and $t : S_1 \in \mathcal{P}$ and $t : S_3 \in \Theta(\mathcal{P})$.

4.4 Non-empty sorts

The definition of models for a specification in G-Algebras doesn't allow for non-empty sort interpretations. But from a practical point of view it is interesting to give a syntactical test of non-emptiness in order to make this condition on models of a specification superfluous. We restrict ourselves in the rest of this paper to specifications with syntactically non-empty sorts, a notion that is defined as follows:

Definition 4.17 *Let $(\Sigma, \mathcal{P})$ with $\Sigma = (\mathcal{S}, \mathcal{F})$ be a specification. A sort $A \in \mathcal{S}$ is called* syntactically empty, *if there is no ground term $t \in T(\mathcal{F})$, s.t. $\mathcal{P} \overset{*}{\vdash}_{DedMem} t : A$, where* **DedMem** *is the set of rules* **Globality**, **MeSubstitutivity**, *else A is called* syntactically non-empty. *$(\Sigma, \mathcal{P})$ has* syntactically non-empty sorts, *if all $A \in \mathcal{S}$ are syntactically non-empty.*

MarkByDeclaration	$(N \cup \{A\}, M, \mathcal{P} \cup \{t : A\})$ $\Longrightarrow (N, M \cup \{A\}, \mathcal{P})$ if $\forall x \in \mathcal{V}ar(t) \ : \ ((x :: B) \in \mathcal{P} \Rightarrow B \in M)$

Figure 3: Rule to search syntactically empty sorts

Figure 3 gives a decision procedure for the syntactical non-emptiness of sorts, assuming the set of sorts to be finite. Starting with the triple $(\mathcal{S}, \emptyset, \mathcal{P})$ the algorithm obviously terminates with $(\emptyset, \mathcal{S}, \mathcal{P}')$ if all sorts contain at least one ground term. Else, the sorts in the first set of the tuple are those without ground terms.

Proposition 4.18 *Let $(\Sigma, \mathcal{P})$ be a specification with $\Sigma = (\mathcal{S}, \mathcal{F})$. The application of the rule* **MarkByDeclaration** *is sound, complete and terminates when applied in a saturating way starting with $(\mathcal{S}, \emptyset, \mathcal{P})$, i.e. if $(M, N, \mathcal{P}')$ is the triple obtained after termination, then the sorts in M are syntactically empty and those in N are syntactically non-empty.*

Proof: Termination is obvious, since $\mathcal{S}$ is finite and the first member of the triple is strictly reduced at each step of the application. In the rest of the proof we refer to the first triple member with M, the second with N and the third with $\mathcal{P}$.

The soundness is shown by induction over the number of rule applications. If a sort A is in M after the first step, then there must be a constant declaration $a : A$ in $\mathcal{P}$, because M was empty in the beginning and so no variable can occur in the used membership declaration. Assuming that there is a membership proof $\Phi_i \vdash t_i : A_i$ for any $A_i \in M$, we take $\bigcup_{j \in J}\{x_j :: A_j \mapsto t_j\}$ as substitution σ for the application of **MeSubstitutivity** to $t : A$ as a ground term membership proof implying the non-emptiness of A when **MarkByDeclaration** is applied to $t : A$ with $\mathcal{V}ar(t) = \{x_j \mid j \in J\}$. Remark, that the non-linearity of term declarations doesn't pose any problem here. Hence any sequence of **MarkByDeclaration**-rule applications corresponds with a ground term membership proof in **DedMem**.

The rule's completeness is also obvious, since any membership proof for a ground term t based on the rules **Globality** and **MeSubstitutivity** can be transformed to a sequence of **MarkByDeclaration**-rule applications, where **Globality** applied to $t' : A$ is replaced by **MarkByDeclaration** applied to $x : \Omega$ for some x with $x :: A$ in $\mathcal{P}$ – such a variable x must exist in any presentation taken under consideration here, since any sort is a subsort of Ω – and **MeSubstitutivity** is simply replaced by the equivalent **MarkByDeclaration** application. Starting from the leaves of the proof with **DedMem** upwards, it can happen that **MarkByDeclaration** isn't applicable anymore, because the sort A used in a membership declaration is already in M or the membership declaration is not in $\mathcal{P}$. But in this case there was already a ground term membership proof for A, because of the soundness, and therefore A is in M if and only if there is a ground term membership proof using **DedMember**. □

4.5 Sort completion

To compute the unifier of two variables in a sort inheriting presentation, we need an extended sort structure $(\mathcal{S}_\diamond, \leq^{syn}_{\mathcal{S}_\diamond})$ containing new sorts representing sort intersections. The transformation is very close to the one in [SS87], which performs the conversion of arbitrary signatures with flat, linear term declarations into regular ones. However, we do not have a restriction to such term declarations, neither

do we have syntactical sorts. Since typing in G-algebras does not only depend on term declarations, but also on equalities, we cannot precompute the exact contents of sorts representing the intersection.

But, in order to avoid problems with empty sorts, as stated in [GM88, SNGM89], we only add new sorts, which surely are non-empty. This can be granted by the condition that there is at least one sort in $\mathcal{S}$ below the new sort. Finally, we want the number of new sorts to be minimal. Unfortunately, the number may still be exponential, since the construction ranges over the power set $2^{\mathcal{S}}$.

Definition 4.19 *Under the assumption 4.6 we can complete the poset* $(\mathcal{S}, \leq_{\mathcal{S}}^{syn})$ *into* $(\mathcal{S}_\Diamond, \leq_{\mathcal{S}_\Diamond}^{syn})$ *in the following way:*

1. $\forall A \in \mathcal{S}$, $\langle A \rangle \in \mathcal{S}_\Diamond$,
2. $\forall S \in 2^{\mathcal{S}}$, $\langle S \rangle \in \mathcal{S}_\Diamond$ *provided:*
   (a) $mlb(S) \neq \emptyset$,
   (b) *all elements in* S *are incomparable w.r.t.* $\leq_{\mathcal{S}}^{syn}$ *in* $(\mathcal{S}, \leq_{\mathcal{S}}^{syn})$,
3. $\langle A \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle B \rangle$ *if* $A \leq_{\mathcal{S}}^{syn} B$,
4. $\langle A \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle S \rangle$ *if* $\langle S \rangle \in \mathcal{S}_\Diamond$, and $A \in mlb(S)$,
5. $\langle S \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle S' \rangle$ *if* $S \subseteq S'$.

Since $(\mathcal{S}_\Diamond, \leq_{\mathcal{S}_\Diamond}^{syn})$ *is a natural extension of* $(\mathcal{S}, \leq_{\mathcal{S}}^{syn})$, *we identify* A *and* $\langle A \rangle$ *for any* $A \in \mathcal{S}$ *and call* $\mathcal{S}_\Diamond$ *and* $(\mathcal{S}_\Diamond, \leq_{\mathcal{S}_\Diamond}^{syn})$ *the* inherited sorts *and* inherited sort structure *of* $(\mathcal{S}, \leq_{\mathcal{S}}^{syn})$. *In order to simplify notations when* S *is given in extension i.e.* $S = \{A, B, \ldots\}$, *we denote* $\langle S \rangle$ *by* $\langle A, B, \ldots \rangle$.

Remark that we do not want to speak of intersection sorts here, since $\mathcal{S}_\Diamond$ does not contain all $\langle S \rangle$ with $S \in \mathcal{S}$, due to the possible emptiness of such sorts.

Example 4.20 *Let* $\mathcal{S} = \{A, B, C, D, E\}$ *with*

$$A \leq_{\mathcal{S}}^{syn} B \leq_{\mathcal{S}}^{syn} C,$$
$$A \leq_{\mathcal{S}}^{syn} D,$$

be the initial sort structure. Then, the associated sort inheriting sort structure is $\mathcal{S}_\Diamond = \{\langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle, \langle E \rangle, \langle B, D \rangle\}$ *together with:*

$$\langle A \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle B \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle C \rangle,$$
$$\langle A \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle D \rangle,$$
$$\langle B, D \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle B \rangle,$$
$$\langle B, D \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle D \rangle,$$
$$\langle A \rangle \leq_{\mathcal{S}_\Diamond}^{syn} \langle B, D \rangle.$$

Clearly, the construction also yields the uniqueness of maximal common subsorts for a set of sorts $S \subseteq \mathcal{S}_\Diamond$.

5 Decorated terms

The undecidability of typing in G-algebra and its dynamic behavior lead us to adopt a specific term structure with a kind of memorizing technique for intermediate results of membership proofs. Furthermore the equality of two terms and the membership of a term to a sort are two sources of information that should be treated and maintained in a parallel way, because the corresponding formulas have a proper separated status in G-algebras, although they are of course interdependent. This is taken into account by the particular data structure, called decorated terms. As a matter of fact, the reader familiar with deduction with constraints will notice that the decorations are actually membership constraints spread in the term and defining a kind of local constraints.

5.1 The Term Structure

Given a G-algebra signature $\Sigma = (\mathcal{S}, \mathcal{F})$ and $(\mathcal{S}_\Diamond, \leq_{\mathcal{S}_\Diamond}^{syn})$ its inherited sort structure, a $\mathcal{S}_\Diamond$-sorted variable set $\mathcal{X}_\Diamond$ and a set of set-variables $\mathbf{V}$ disjoint from $\mathcal{X}_\Diamond$, we define as follows the decorated $(\Sigma, \mathcal{X}_\Diamond)$-terms.

Definition 5.1 *A* decoration *is either a subset of $\mathcal{S}_\Diamond$, called ground decoration, or the union of a ground decoration and a variable s in $\mathbf{V}$, representing a ground decoration. Then, a decorated $(\Sigma, \mathcal{X}_\Diamond)$-term, or* decorated term *for short if $(\Sigma, \mathcal{X}_\Diamond)$ is clear from the context, is:*

1. *either a pair (x, S) of a variable $x \in \mathcal{X}_\Diamond$ and a decoration S, written $x^{:S}$,*
2. *or of the form $(f, S)((t_1, S_1), \ldots, (t_n, S_n))$, if $(t_1, S_1), \ldots, (t_n, S_n)$ are decorated $(\Sigma, \mathcal{X}_\Diamond)$-terms, $f \in \mathcal{F}$ with $arity(f) = n$ and S is a decoration. Such a term is written $f(t_1{}^{:S_1}, \ldots, t_n{}^{:S_n})^{:S}$.*

$\mathcal{V}ar(t^{:S})$ stands for the set of variables without their decorations contained in $t^{:S}$.

$\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ is the set of decorated $(\Sigma, \mathcal{X}_\Diamond)$-terms, $\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F})$ the set of ground $(\Sigma, \emptyset)$-terms. For any decorated term or in general any formula ν involving decorated terms, the set $\mathcal{V}ar_d(\nu)$ is the set of all decorated variables including their decorations in ν, considering the same variable with different decorations as different elements. In order to precise the difference between $\mathcal{V}ar(\nu)$ and $\mathcal{V}ar_d(\nu)$, we give an example:

Example 5.2 *Let $\mathcal{T} = \{x^{:S}, f(y^{:U}), g(a), h(x^{:T})\}$. Then, we have:*

$$\begin{aligned} \mathcal{V}ar(\mathcal{T}) &= \{x, y\} \\ \mathcal{V}ar_d(\mathcal{T}) &= \{x^{:S}, y^{:U}, x^{:T}\} \end{aligned}$$

Now we can extend the classical notions of undecorated terms to decorated terms.

Definition 5.3 *Let $t^{:S}$ be a decorated term in $\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$.*
The theory of occurrences, written $\mathcal{O}cc$, is the monoid $(\mathbf{N} \cup \{\Lambda\}, \cdot)$, where Λ denotes the neutral element.
The occurrences of $t^{:S}$, written $\mathcal{O}cc(t^{:S})$, with $\mathcal{O}cc(t^{:S}) \subseteq \mathcal{O}cc$, are defined as follows:

$$\begin{aligned} \mathcal{O}cc(x^{:S}) &= \{\Lambda\} && \text{if } x \in \mathcal{X}_\Diamond \\ \mathcal{O}cc(f(t_1, \ldots, t_n)^{:S}) &= \{\Lambda\} \cup_{i \in [1..n]} \{i.\omega \mid \omega \in \mathcal{O}cc(t_i)\} && \text{otherwise.} \end{aligned}$$

Let $\omega, \upsilon \in \mathcal{O}cc$. The occurrence prefix ordering $\preceq$ *is the smallest binary relation over occurrences, s.t. $\omega \preceq \upsilon$ if there exists some $\nu \in \mathcal{O}cc$ with $\upsilon = \omega.\nu$.*

The lexicographic occurrence ordering $\preceq_{lex}$ *is the smallest binary relation over occurrences, s.t.* $\omega \preceq_{lex} \upsilon$ *if there exist* $\nu, \nu_1, \nu_2 \in \mathcal{O}cc$ *and* $m, n \in \mathbb{N}$, *s.t.* $\omega = \nu.m.\nu_1$, $\upsilon = \nu.n.\nu_2$ *and* $m < n$.
The term projection function $._{|} : \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond) \times \mathcal{O}cc \mapsto \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, *is defined as follows:*

$$\begin{array}{lcll} t^{:S}{}_{|_\Lambda} & = & t^{:S} & \\ f(t_1, \ldots, t_n)^{:S}{}_{|_{i.\omega}} & = & t_i{}_{|_\omega} & \text{if } i \in [1..n] \\ t^{:S}{}_{|_\omega} & = & \bot & \text{otherwise} \end{array}$$

The variable occurrences *of* $t^{:S}$, *written* $\mathcal{VO}cc(t^{:S})$, *are defined as* $\{\omega \in \mathcal{O}cc(t^{:S}) \mid t^{:S}{}_{|_\omega} \in \mathcal{X}_\Diamond\}$.
The non-variable occurrences *of* $t^{:S}$, *written* $\mathcal{NVO}cc(t^{:S})$, *are those in* $\mathcal{O}cc(t^{:S}) \setminus \mathcal{VO}cc(t^{:S})$.
The decoration S *of a term* $t^{:S}$ *is denoted by* $Deco(t^{:S})$.

The term $(t^{:S})_{nd}$ is $t^{:S}$ without its decoration, i.e. a term in $\mathcal{T}((\mathcal{S}_\Diamond, \mathcal{F}), \mathcal{X}_\Diamond)$.

Instead of $t^{:\{A,B\}}$, we may also write $t^{:A,B}$ and, when the top decoration does not matter, we also omit the exponent, in order to simplify the syntax.

A undecorated term t is a $\mathcal{T}(\Sigma, \mathcal{X})$-*instance* of $x^{:S} \in \mathcal{X}_\Diamond$, if $t \in \mathcal{T}(\Sigma, \mathcal{X})$ and if $(x :: \langle A_i \rangle_{i=,\ldots,n})$, then $\forall i = 1, \ldots, n, \mathcal{P} \vdash (\alpha(x) : \langle A_i \rangle)$. In this case the homomorphism generated by $(x \mapsto t)$ is called a $\mathcal{T}(\Sigma, \mathcal{X})$-*assignment* of $x^{:S}$, in the term algebra $\mathcal{T}((\mathcal{S}_\Diamond, \mathcal{F}), \mathcal{X}_\Diamond)$.

This extends canonically to sets of decorated variables and decorated terms, where the condition must hold for all decorated variables in $t^{:S}$. The set of $\mathcal{T}(\Sigma, \mathcal{X})$-assignments of a decorated term $t^{:S}$ is written $\mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S})$.

The extraction of the formula part present in a decorated term is defined as follows:

Definition 5.4 *The* decoration formulas *associated with a decorated term* t, *noted* $\mathcal{P}_t$, *are the set*

$$\{u' : B \mid \begin{array}{l} \exists u^{:U} \in subterm\_set(t), \exists \langle B, \ldots \rangle \in U \text{ and} \\ u' \text{ is a } \mathcal{T}(\Sigma, \mathcal{X}) - \text{instance of } u^{:U}\} \end{array}$$

A decorated $(\Sigma, \mathcal{X}_\Diamond)$-*term* $t^{:S}$ *is* valid *in a presentation* $\mathcal{P}$, *if* $\mathcal{P}_{t^{:S}} \subseteq Th(\mathcal{P})$. $Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ *and* $Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F})$ *are the set of all valid, decorated terms, and valid, decorated ground terms, respectively.*

Hence, valid decorated terms are a combined representation of terms and true membership formulas. Obviously, all subterms of a valid term are also valid, since $\mathcal{P}_u \subseteq \mathcal{P}_{t[u]_\omega}$ for all $\omega \in \mathcal{O}cc(t)$.

Each term t in $\mathcal{T}(\Sigma, \mathcal{X})$ is identified with the decorated term $t^{:\downarrow\emptyset}$ with an empty set of sorts at each node except the variable positions, where the decoration contains the sort of the variable only. Notice that the definition of valid terms allows for empty decorations, i.e. every $t^{:\downarrow\emptyset} \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ is valid.

This notation also extends to sets, i.e. $\{t^{:\downarrow\emptyset} \mid t \in \mathcal{T}\}$ is written $\mathcal{T}^{:\downarrow\emptyset}$ for any $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$.

All other notions concerning decorated terms are defined in the same way as for classical terms. The top occurrence in decorated terms is denoted by Λ. The equality over decorated terms, noted $=_d$, is the conjunction of classical equality over variables and function symbols with the set equality over the corresponding decorations. The negation of $=_d$ is denoted $\neq_d$. $t^{:S} =_{nd} t'^{:S'}$ stands for $(t^{:S})_{nd} = (t'^{:S'})_{nd}$.

Several decorations may seem to be in conflict when we write for instance: $\forall \omega \in \mathcal{O}cc(t^{:S}) : (t^{:S}{}_{|_\omega})^{:S'} =_d t'$ in the case where $\omega = \Lambda$. Then the actual decoration of t' is always the outermost mentioned, i.e. in this case S' and not S. The only exception is of course the replacement of subterms, i.e. the decoration of $t^{:S}[u^{:S'}]_\omega$ at occurrence ω is clearly S' and not the one of $t^{:S}{}_{|_\omega}$.

Furthermore, we need to adapt the notion of sort inheritance on decorated terms. The definition given now is relative to a subset of valid decorated terms.

Definition 5.5 *Let $\mathcal{T} \subseteq Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ and $\leq$ be a subsort relation. A specification $((\mathcal{S}, \mathcal{F}), \mathcal{P})$ is* $\mathcal{T}$-sort inheriting *with respect to $\leq$, if:*

$$\forall t^{:\mathcal{T}} \in \mathcal{T} \ : \ A, B \in T \Rightarrow (\exists C \in \mathcal{S} \ : \ C \leq A, B).$$

Clearly, $Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sort inheritance w.r.t. $\leq_{\mathcal{S}}^{syn}$ is equivalent to sort inheritance of the specification. Consequently, we can write "sort inheritance" instead of "$Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sort inheritance w.r.t. $\leq_{\mathcal{S}}^{syn}$". As an immediate consequence of this definition and since $\leq_{\mathcal{S}}^{syn} \subseteq \leq_{\mathcal{S}}^{sem}$, the following implication holds:

Lemma 5.6 *If $((\mathcal{S}, \mathcal{F}), \mathcal{P})$ is $\mathcal{T}$-sort inheriting w.r.t. $\leq_{\mathcal{S}}^{syn}$, then it is $\mathcal{T}$-sort inheriting w.r.t. $\leq_{\mathcal{S}}^{sem}$.*

In the rest of the paper, we mean implicitely $\mathcal{T}$-sort inheritance w.r.t. $\leq_{\mathcal{S}}^{syn}$, when we use $\mathcal{T}$-sort inheritance. The motivation for the following definition is to generalize the fact that when a term t belongs to a sort A, it also belongs to all sorts greater than A.

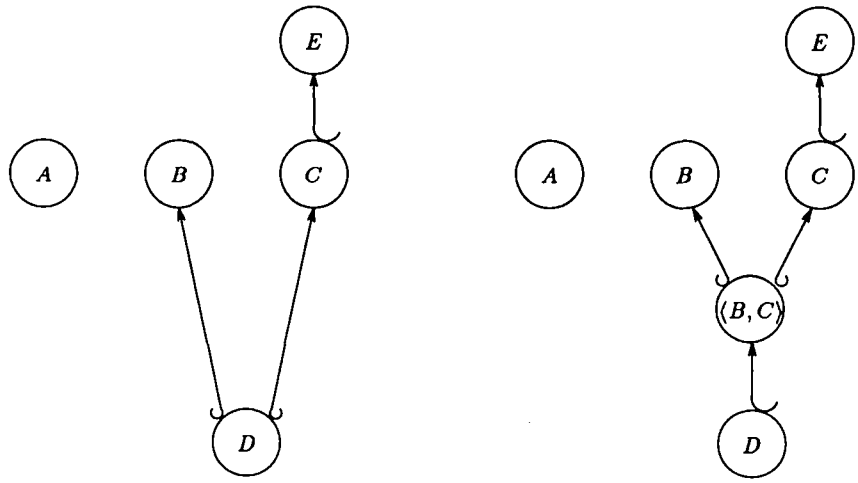
Definition 5.7 *Let $(\Sigma, \mathcal{P})$ be a specification with $\Sigma = (\mathcal{S}, \mathcal{F})$, $S \subseteq \mathcal{S}_\Diamond$ and $t^{:S} \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. The* sort inheritance closure *of S, written $\widehat{S}$ is the set:*

$$\widehat{S} = \{D \in \mathcal{S}_\Diamond \mid \exists \langle T \rangle, \langle T' \rangle \in S \ : \ \langle T \cup T' \rangle \in \mathcal{S}_\Diamond \text{ and } \langle T \cup T' \rangle \leq_{\mathcal{S}_\Diamond}^{syn} D\}.$$

The sort inheritance closure *of $t^{:S}$ is the term $t'^{:S'}$ satisfying $t^{:S} =_{nd} t'^{:S'}$ and $\forall \omega \in Occ(t^{:S}) \ : \ Deco(t'^{:S'}|_\omega) = \widehat{Deco(t^{:S}|_\omega)}$.*

We illustrate this with an example:

Example 5.8 *Let $\mathcal{S} = \{A, B, C, D, E\}$ with $C \leq_{\mathcal{S}}^{syn} E$, $D \leq_{\mathcal{S}}^{syn} B, C$ and therefore $\mathcal{S}_\Diamond = \{A, B, C, D, E, \langle B, C \rangle\}$ with $C \leq_{\mathcal{S}_\Diamond}^{syn} E$, $D \leq_{\mathcal{S}_\Diamond}^{syn} \langle B, C \rangle$ and $\langle B, C \rangle \leq_{\mathcal{S}_\Diamond}^{syn} B, C$:*



Let furthermore $S_1 = \{A\}$, $S_2 = \{B, C\}$ and $S_3 = \{A, B, C\}$ be subsets of $\mathcal{S}_\Diamond$.

Then $\widehat{S_1} = \{A\}$, $\widehat{S_2} = \{B, C, \langle B, C \rangle, E\}$ and $\widehat{S_3} = \{A, B, C, \langle B, C \rangle, E\}$.

If $S = \{A\}$, we write $\hat{A}$ instead of $\hat{S}$. For finite sort sets S, S', the notation $S \lesssim S'$ stands for $\widehat{S} \subseteq \widehat{S'}$ and $\not\lesssim$ for its negation. If $S \lesssim S'$ and $S' \lesssim S$, we write $S \approx S'$. Furthermore, $t^{:S} \cong_d t'^{:S'}$ stands for $\widehat{t^{:S}} = \widehat{t'^{:S'}}$. When $\mathcal{T}$ and $\mathcal{T}'$ are sets of decorated terms, then $\mathcal{T} \cong_d \mathcal{T}'$ if $\{\hat{t} \mid t \in \mathcal{T}\} = \{\hat{t'} \mid t' \in \mathcal{T}'\}$. Our general assumption 4.6 gives us the decidability of these relations.

Proposition 5.9 *Given finite sort sets $S, S' \subseteq \mathcal{S}_\emptyset$, it is decidable if $S \lesssim S'$.*

Proof: Let T be a subset of $\mathcal{S}$ (resp. $\mathcal{S}_\emptyset$) and $min(T) = \{C \mid \nexists D \in T \ : \ D <_{\mathcal{S}}^{syn} C\}$ (resp. $min(T) = \{C \mid \nexists D \in T \ : \ D <_{\mathcal{S}_\emptyset}^{syn} C\}$). Clearly, $min(T)$ has to be unique, since assuming w.l.o.g. another set $\overline{T}$ with the same property leads to a $C \in min(T) \setminus \overline{T}$, s.t. there is a $\overline{C} \in \overline{T}$ with $\overline{C} <_{\mathcal{S}}^{syn} C$, which is in contradiction to $C \in min(T)$, knowing that $\overline{T} \subseteq T$. This minimization step is done in order to simplify the construction of sorts in $\mathcal{S}_\emptyset$, needed for $\widehat{S}$ and $\widehat{S'}$.

Now, we define ∇ to be the operator that adds $\langle min(T \cup T') \rangle$ to a sort set M for all incomparable sorts $\langle T \rangle, \langle T' \rangle \in M$, if $\langle min(T \cup T') \rangle \in \mathcal{S}_\emptyset$. Clearly, the size of $\langle min(T \cup T') \rangle$ is limited by the sum of the size of all sorts in M. Therefore, the limit $\nabla^\infty(M)$ is reached after a finite number of steps, giving us the computability of $U = \nabla^\infty(S)$ and $U' = \nabla^\infty(S')$. U and U' are unique since $min()$ is and ∇ always adds sorts, but never erases them.

Consequently, $S \lesssim S'$ iff for all $C' \in min(U')$ there is some $C \in min(U)$ with $C \leq_{\mathcal{S}_\emptyset}^{syn} C'$. □

Corollary 5.10 *Let S, S' be finite sort sets and $t^{:S}, t^{:S'}$ be decorated terms. Then the relations $S \not\lesssim S'$, $S \approx S'$ and $t^{:S} \cong_d t^{:S'}$ are decidable.*

5.2 Subsumption for Decorated Terms

Let us extend the classical term subsumption on decorated terms.

Definition 5.11 *Let $(\Sigma, \mathcal{P})$ be a subsort unique specification with $\Sigma = (\mathcal{S}, \mathcal{F})$ and $t^{:S}, t'^{:S'}$ be valid decorated terms. Then $t'^{:S'}$ is* subsumed modulo sort inheritance *by $t^{:S}$, written $t^{:S} \lesssim_d t'^{:S'}$, if:*

1. $\mathcal{O}cc(t^{:S}) \subseteq \mathcal{O}cc(t'^{:S'})$,
2. $\forall \omega \in \mathcal{NV}Occ(t^{:S}) \ : \ (t^{:\downarrow\emptyset})(\omega) = (t'^{:\downarrow\emptyset})(\omega)$,
3. $\forall \omega \in \mathcal{NV}Occ(t^{:S}) \ : \ Deco(t^{:S}{}_{|\omega}) \approx Deco(t'^{:S'}{}_{|\omega})$ *and*
4. $\forall x \in \mathcal{V}ar(t^{:S}) \ : \ \exists t_x^{:T_x} \ : \ \forall \omega \in \mathcal{O}cc(t^{:S}) \ : \ (t^{:S}{}_{|\omega} \cong_d x^{:S_x} \Rightarrow (t'^{:S'}{}_{|\omega} \cong_d t_x^{:T_x}$ *and* $S_x \lesssim T_x))$.

The last condition guarantees that each variable is bound to a unique decorated term modulo sort inheritance. Clearly, if $t^{:S} \lesssim_d t'^{:S'}$ and $t'^{:S'} \lesssim_d t^{:S}$, then $t^{:S} \cong_d t'^{:S'}$ modulo variable renaming. Note that $\lesssim_d$ stands for $\lesssim_d$ without $\lesssim_d \cap \gtrsim_d$.

Example 5.12 *Let $\mathcal{S}_\emptyset = \{A, B, C\}$ and $\leq_{\mathcal{S}}^{syn} = \emptyset$.*
Assuming all terms in the following be valid, we get

$$f(x^{:\{A\}}, x^{:\{A\}}, y^{:\{B\}})^{:\{C\}} \lesssim_d f(a^{:\{A\}}, a^{:\{A\}}, b^{:\{B\}})^{:\{C\}},$$

but neither
$$f(x^{:\{A\}}, x^{:\{A\}}, y^{:\{B\}})^{:\{C\}} \lesssim_d f(a^{:\{B\}}, a^{:\{A\}}, b^{:\{B\}})^{:\{C\}}$$

nor
$$f(x^{:\{A\}}, x^{:\{A\}}, y^{:\{B\}})^{:\{C\}} \lesssim_d f(a^{:\{A,B\}}, a^{:\{A\}}, b^{:\{B\}})^{:\{C\}}$$

or
$$f(x^{:\{A\}}, x^{:\{A\}}, y^{:\{B\}})^{:\{C\}} \lesssim_d f(b^{:\{A\}}, a^{:\{A\}}, b^{:\{B\}})^{:\{C\}} \quad \textit{holds.}$$

5.3 Substitutions

Decorated substitutions are a subset of the classical well-defined order-sorted substitutions. As already mentioned, we restrict the used membership theory to the information already existing in the term nodes.

Definition 5.13 *A* decorated substitution σ *is a function replacing decorated variables in* $\mathcal{X}_\Diamond$ *by decorated terms, such that if* $\sigma(x^{:S}) =_d t^{:T}$ *with* $t^{:T} \neq x^{:S}$ *and* $(x :: A) \in \mathcal{P}$, *then* $A \in S$, $A \in \widehat{T}$ *and* $t^{:T}$ *is valid.* σ *is represented by its graph* $\bigcup_{i\in[1..n]}\{x_i^{:S_i} \mapsto t_i^{:T_i}\}$ *with* $\mathcal{D}om(\sigma) = \{x_i^{:S_i} \mid i \in [1..n]\}$, $\mathcal{I}m(\sigma) = \{t_i^{:S_i} \mid i \in [1..n]\}$, $\mathcal{R}an(\sigma) = \mathcal{V}ar_d(\mathcal{I}m(\sigma))$. *Decorated substitutions can be extended to homomorphisms over decorated terms as follows:*

$$\begin{array}{rcll} \sigma(x^{:S_0}) & =_d & t^{:T} & \text{if } (x^{:S} \mapsto t^{:T}) \in \sigma,\ x :: A, \text{ and } \widehat{A} \approx S_0 \\ \sigma(y^{:S}) & =_d & y^{:S} & \text{if } y \in \mathcal{X}_\Diamond \text{ and } y \notin \mathcal{D}om(\sigma) \\ \sigma(f(t_1,\ldots,t_n)) & =_d & f(\sigma(t_1),\ldots,\sigma(t_n)) & \text{otherwise.} \end{array}$$

Let $\mathcal{T}, \mathcal{T}_X \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. $SUBST$ *denotes the set of all decorated substitutions. The set of all* decorated substitutions of $\mathcal{T}_X$ into $\mathcal{T}$, *written* $SUBST_{|_{\mathcal{T}_X \to \mathcal{T}}}$, *is the set* $\Sigma \subseteq SUBST$ *with* $\forall \sigma \in \Sigma, \forall t^{:T} \in \mathcal{T}_X \ :\ \sigma(t^{:T}) \in \mathcal{T}$.

The concatenation of two decorated substitutions σ, τ, *written* $\sigma \circ \tau$, *is the composition of the corresponding functions* ($\sigma \circ \tau(t) = \sigma(\tau(t))$ *for any term* t).

It is important to notice that we do not try to compute the lowest sort of the term in the image of σ, in order to test if this term belongs to a subsort of the variable. Instead, we simply decide this using the sorts in the top decoration of the term modulo sort inheritance.

Lemma 5.14 *If* t *is a valid decorated term and* σ *a decorated substitution, then* $\sigma(t)$ *is a valid decorated term.*

Proof: This is an easy consequence of the application of the deduction rule **MeSubstitutivity**. □

Corollary 5.15 *The composition* $\sigma \circ \tau$ *of two decorated substitutions*

$$\sigma = \{x_i^{:S_i} \mapsto t_i^{:T_i}\}_{i\in I} \text{ and } \tau = \{x_j'^{:S_j'} \mapsto t_j'^{:T_j'}\}_{j\in J},$$

is a decorated substitution, provided that $\forall j \in J \ :\ \forall i \in I \ :\ x_i^{:U} \in \mathcal{V}ar_d(t_j'^{:T_j'}) \Rightarrow U \lneq S_i$.

Corollary 5.16 *Let* σ, τ *be two decorated substitutions with* $\mathcal{D}om(\sigma) \cap \mathcal{R}an(\tau) = \emptyset$. *Then the composition* $\sigma \circ \tau$ *is also a decorated substitution.*

Equality and orderings over decorated substitutions are essentially the same as in the classical order-sorted case.

Definition 5.17 *Let* σ *and* τ *be two decorated substitutions,* $\mathcal{V}$ *a finite decorated variable set.* σ is more general than τ over $\mathcal{V}$, *written* $\sigma \lesssim_d^{\mathcal{V}} \tau$, *if* $\forall t \in \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ *with* $\mathcal{V}ar_d(t) \subseteq \mathcal{V} \ :\ \sigma(t) \lesssim_d \tau(t)$. *Then we say that* σ is equal to τ over $\mathcal{V}$, *written* $\sigma \cong_d^{\mathcal{V}} \tau$, *if* $\sigma(x^{:S}) \cong_d \tau(x^{:S})$ *for all* $x^{:S} \in \mathcal{V}$.

Of course, we get the classical equivalence of matching and term subsumption whose proof can be found in [HKK93].

Proposition 5.18 *Let* $t^{:S}, t'^{:S'} \in \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. *Then* $\exists \sigma \in SUBST \ :\ \sigma(t) \cong_d t'$ *is equivalent to* $t \lesssim_d t'$.

Proof: $\Rightarrow$: Take $t_x^{:T_x} = \sigma(x^{:S_x})$ for all $x^{:S_x} \in \mathcal{V}ar_d(t^{:S})$.

$\Leftarrow$: Define σ as $\{x^{:S_x} \mapsto \widehat{t_x^{:T_x}}\}_{x^{:S_x}\in\mathcal{V}ar_d(t^{:S})}$. □

As expected, it is sufficient to find a complementary substitution in order to prove subsumption. However, this is not a necessary condition.

Proposition 5.19 *Let $(\Sigma, \mathcal{P})$ be a specification, σ, τ be two decorated substitutions and $\mathcal{V}$ a finite set of decorated variables. Then:*

$$\exists \rho \ : \ \rho \circ \sigma \cong_d^{\mathcal{V}} \tau \Rightarrow \sigma \lesssim_d^{\mathcal{V}} \tau.$$

Proof: Assume $t \in \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ with $\mathcal{V}ar_d(t) \subseteq \mathcal{V}$. We have $\rho \circ \sigma(t) \cong_d \tau(t)$. Take $t_x{}^{:T_x} =_d \rho(x^{:\{\emptyset\}})$ for any $x \in \mathcal{V}ar(\sigma(t))$. Then $\sigma \lesssim_d^{\mathcal{V}} \tau$ is an immediate consequence of definition 5.11. □

The other direction is not always true, as the following example shows:

Example 5.20 *Let $\Sigma = (\{\Omega\}, \{f, a, b\})$, s.t. $arity(f) = 1$, $arity(a) = arity(b) = 0$ and $\mathcal{V} = \{x :: \Omega, y :: \Omega\}$. Then $\sigma = \{x \mapsto f(z^{:\{\Omega\}})^{:\Omega}, y \mapsto f(z^{:\{\Omega\}})^{:\Omega}\}$ subsumes $\tau = \{x \mapsto f(a^{:\{\Omega\}})^{:\Omega}, y \mapsto f(b^{:\{\Omega\}})^{:\Omega}\}$ over $\mathcal{V}$, since there is no term containing x and y at the same time, but there is no ρ, s.t. $\rho \circ \sigma \cong_d^{\mathcal{V}} \tau$.*

However, we get the following result adapted from [Hue76]:

Proposition 5.21 *Let $(\Sigma, \mathcal{P})$ be a specification containing some non-monadic function symbol defined over the universe for each argument, σ, τ be two decorated substitutions and $\mathcal{V}$ a finite set of decorated variables. Then:*

$$\sigma \lesssim_d^{\mathcal{V}} \tau \Leftrightarrow \exists \rho \ : \ \rho \circ \sigma \cong_d^{\mathcal{V}} \tau.$$

Proof: The fact that Σ contains some non-monadic function symbol defined over the universe for each argument guarantees the existence of a n-ary symbol f, $n > 1$, that allows us to construct some term t containing all variables in $\mathcal{V}$. By the assumption $\sigma \lesssim_d^{\mathcal{V}} \tau$, we get $\sigma(t) \lesssim_d \tau(t)$, i.e. there is a $t_x{}^{:T_x}$ with $\forall \omega \in \mathcal{VO}cc(\sigma(t)) \ : \ \sigma(t)_{|_\omega} = x^{:S'} \Rightarrow \tau(t)_{|_\omega} \cong_d t_x{}^{:T_x}$. Therefore we can take:

$$\rho = \{x^{:S_x} \mapsto t_x{}^{:T_x} \mid \exists S' : x^{:S'} \in \mathcal{V} \text{ and } S_x = \bigcup_{x^{:S} \in \mathcal{V}ar_d(\sigma(t))} S\}.$$

Clearly, ρ and $\rho \circ \sigma$ are decorated substitutions and $t_x{}^{:T_x}$ is a valid term, since all terms in $\mathcal{I}m(\tau)$ are and t is. Therefore, by the construction of t and $t_x{}^{:T_x}$, we get for all $x^{:S} \in \mathcal{R}an(\sigma) \ : \ \rho(x^{:S}) \cong_d \tau(x^{:\{\emptyset\}})$. Consequently, there is a ρ with $\rho \circ \sigma(t) \cong_d \tau(t)$ for all $t \in \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ with $\mathcal{V}ar_d(t) \subseteq \mathcal{V}$. □

Anyway, given a valid term t, computing a substitution ρ with $\rho \circ \tau(t) \cong_d \sigma(t)$ results in solving a matching problem, if $\tau \lesssim_d^{\mathcal{V}ar(t)} \sigma$. This is a consequence of $\tau(t) \lesssim_d \sigma(t)$ and Proposition 5.18.

6 Strict Decorated Matching and Unification

Algorithms for matching and unification on decorated terms are designed in this section. In the following, we assume all terms in the input problems of the algorithms to be valid. Matching and unification are called strict here because they require to take into account at each node both the identity of function symbols and the equality of decorations modulo the sort inheritance closure.

Matching and unification procedure presented in this paper are described as set of transformation rules working over terms representing the problem. A transformation rule is *sound* if the set of solutions of the resulting problem is subsumed by the one of the initial problem. If the set of solutions of the resulting problem subsumes the solutions of the initial problem, then the transformation rule is *complete*. The transformation rules have the following form:

$$t \Longrightarrow t' \text{ if } condition$$

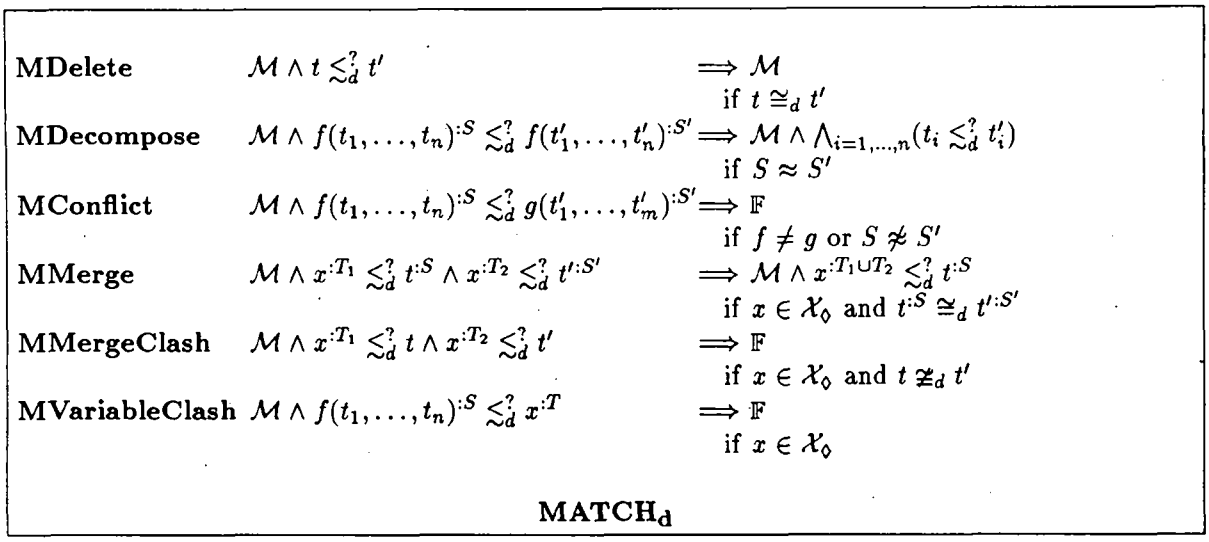


Figure 4: Rules for strict, syntactic, decorated matching

where t and t' are terms representing problems and *condition* is a boolean expression that controls application of the rule, namely the rule is applicable to a problem s if and only if (iff for short) there is a substitution σ with $\sigma(t) = s$ and $\sigma(\textit{condition})$ is satisfied. Such a procedure tries to apply its rules deterministically, i.e. without backtracking, to the current problem as long as possible, replacing the current problem by $\sigma(t')$. Consequently, the procedure terminates if no more rule is applicable and in this case we call the procedure an *algorithm.*

6.1 A Restricted Version of Semantical Order-Sorted Matching

Let us first clarify the basic notions. If t and t' are decorated terms then a *(decorated) matching equation* has the form $t \lesssim_d^? t'$. A *(decorated) matching problem* $\mathcal{M}$ is either a finite conjunction of matching equations, or a new symbol $\mathbb{T}$ denoting an empty conjunction ($\bigwedge_{i\in\emptyset} \equiv \mathbb{T}$), or $\mathbb{F}$, a new symbol denoting an unsatisfiable problem. If $\mathcal{M} = \bigwedge_{i\in I}(t_i \lesssim_d^? s_i)$ is a matching problem, then $\mathit{Conj}(\mathcal{M})$ denotes the set $\bigcup_{i\in I}\{t_i \lesssim_d^? s_i\}$. A matching problem $\mathcal{M} = \bigwedge_{i\in I}(s_i \lesssim_d^? t_i)$ is called *variable disjoint* if the variables of the equation's left-hand sides are different from those on the right-hand side. $\mathcal{M}$ is said to be in *solved form* if it is of the form $\mathbb{T}$, $\mathbb{F}$ or $\bigwedge_{i\in I}(x_i :: A_i \lesssim_d^? t_i^{:S_i})$, such that $x_m \neq x_n \in \mathcal{X}_\Diamond$ for $m, n \in I$ with $m \neq n$. Last but not least, let us define the notion of a solution of a matching problem.

Definition 6.1 *A substitution σ is a* strict solution *(or just $\mathcal{D}$-solution or $\mathcal{D}$-match) of the matching problem $\mathcal{M}$, if for each $(t \lesssim_d^? t') \in \mathit{Conj}(\mathcal{M})$, it satisfies $\sigma(t) \cong_d t'$. The set of all $\mathcal{D}$-matches of a problem $\mathcal{M}$ is denoted $\mathit{Sol}(\mathcal{M})$.*

A decorated substitution $\sigma \in \mathit{Sol}(\mathcal{M})$ is called principal solution *of $\mathcal{M}$, if for all decorated substitutions $\tau \in \mathit{Sol}(\mathcal{M})$: $\sigma \lesssim_d^{\mathcal{V}ar(\mathcal{M})} \tau$.*

The rules in Figure 4 show how to compute a unique solved form for a decorated matching problem. Their main characteristic is that the decorations in the two members of a match equation should be compatible at each level (see in particular the rule **MDecompose**).

Proposition 6.2 *[HK92] The rules in* $\mathbf{MATCH_d}$ *are sound, complete and their application terminates for any variable disjoint decorated matching problem $\mathcal{M}$ as input, yielding a unique solved form of $\mathcal{M}$.*

Remains the problem of solution extraction from the found solved form:

Proposition 6.3 *[HK92] Let $\mathcal{M}_{sf}$ be the solved form of a matching problem $\mathcal{M}$ found by normalizing using the rules in* $\mathbf{MATCH_d}$. *If $\mathcal{M}_{sf}$ has the form $\bigwedge_{i\in I}((x_i :: A_i)^{:S_i} \lesssim_d^? t_i^{:T_i})$ such that the conditions $\forall i \in I, A_i \in T_i$ and $S_i \subsetneq T_i$ are satisfied, then the decorated substitution $\sigma = \{x_i^{:S_i} \mapsto t_i^{:T_i}\}_{i\in I}$ is a principal $\mathcal{D}$-solution of $\mathcal{M}$, else there is no $\mathcal{D}$-solution for $\mathcal{M}$.*

Corollary 6.4 *Strict decorated matching is decidable.*

One may wonder how the undecidability of typing in G-algebras is handled here. Clearly, the additional condition on the decorations of a solution brings decidability. It allows using exactly those sort memberships of a term that can be found in its decoration, not necessarily all that are true under the current presentation, as the following example shows:

Example 6.5 *Let $\mathcal{P} = \{a : A, b : B, x :: A, y :: B, x : B, f(x) : A, f(y) : B, a = b\}$.*
For $\mathcal{M}_1 = (f(x^{:\{A\}})^{:\{A\}} \lesssim_d^? f(a^{:\{A,B\}})^{:\{A\}})$, then $\sigma = \{x^{:\{A\}} \mapsto a^{:\{A,B\}}\}$ is the principal solution of $\mathcal{M}_1$ computed by $\mathbf{MATCH_d}$.
Let $\mathcal{M}_2 = (f(x^{:\{A\}})^{:\{A\}} \lesssim_d^? f(a^{:\{A,B\}})^{:\{A,B\}})$, then $Sol(\mathcal{M}_2) = \emptyset$, because there is no way to add the sort B to the decoration of $f(x^{:\{A\}})^{:\{A\}}$.

6.2 Strict Decorated Unification in Sort Inheriting Presentations

Strict decorated unification uses a quite similar formalism. A *(decorated) unification equation* is written $t \cong_d^? t'$, where t, t' are decorated terms. A *(decorated) unification problem* $\mathcal{U}$ is either a finite conjunction of unification equations, or an empty conjunction $\mathbb{T} \equiv \bigwedge_{i\in\emptyset}$, or $\mathbb{F}$ which denotes the unsolvable unification problem. Any unification problem may be preceded by existential quantifiers $\exists x_1, \ldots, \exists x_n$. If $\mathcal{U} = \bigwedge_{i\in I}(t_i \cong_d^? s_i)$ is a unification problem, then $Conj(\mathcal{U})$ denotes the set $\bigcup_{i\in I}\{t_i \cong_d^? s_i\}$.

The set of rules compute a special form of unification problems that facilitates the extraction of solutions, the so-called solved form. We use here a sorted version of the dag solved forms of [JK91].

Definition 6.6 *A unification problem $\mathcal{U}$ is in* solved form, *if it has the following form:*

1. $\mathbb{T}$ *or* $\mathbb{F}$, *or else,*
2. *of the form* $\exists z_1, \ldots z_m \ \bigwedge_{i\in[1..n]} x_i^{:S_i} \cong_d^? t_i^{:T_i}$ *with* $x_i \in \mathcal{X}_\Diamond$, *such that:*
   - *(a)* $\forall 1 \leq i < j \leq n : \ x_i \neq x_j$,
   - *(b)* $\forall 1 \leq i \leq j \leq n : \ x_i \notin \mathcal{V}ar(t_j)$,
   - *(c)* $\forall 1 \leq i \leq n : \ $ if $t_i \in \mathcal{X}_\Diamond$, then $x_i \notin \{z_1, \ldots, z_m\}$,
   - *(d)* $\forall 1 \leq k \leq m : \ \exists 1 \leq j \leq n : \ z_k \in \mathcal{V}ar(t_j)$,
   - *(e)* $\forall 1 \leq i \leq n : \ S_i \subsetneq T_i$.

A solution of a strict unification problem is defined with respect to a subset $\mathcal{T}$ of valid terms, as follows:

Definition 6.7 *Let $\mathcal{P}$ be a presentation, $\mathcal{U}$ be a conjunction of unification equations, $\mathcal{T}_\mathcal{U} = terms(\mathcal{U})$ and $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. A decorated substitution $\sigma \in SUBST_{|\mathcal{T}_\mathcal{U} \to \mathcal{T}}$ is a* strict decorated $\mathcal{T}$-unifier *(w.r.t. $\mathcal{P}$), if:*

$$\forall (t \cong_d^? t') \in Conj(\mathcal{U}), \sigma(t) \cong_d \sigma(t').$$

$SU_\mathcal{P}(\mathcal{U})_{|\mathcal{T}}$, called strict $\mathcal{T}$-unifier set *of $\mathcal{U}$, is the set of all strict decorated $\mathcal{T}$-unifiers of $\mathcal{U}$.*

UDelete	$\exists \vec{z}:\ (\mathcal{U} \wedge t \cong_d^? t')$ $\Longrightarrow \exists \vec{z}:\ (\mathcal{U})$ if $t \cong_d t'$
UDecompose	$\exists \vec{z}:\ (\mathcal{U} \wedge f(t_1,\ldots,t_n)^{:S} \cong_d^? f(t'_1,\ldots,t'_n)^{:S'})$ $\Longrightarrow \exists \vec{z}:\ (\mathcal{U} \wedge \bigwedge_{i\in[1..n]}(t_i \cong_d^? t'_i))$ if $S \approx S'$
UCoalesce	$\exists \vec{z}:\ (\mathcal{U} \wedge x^{:S} \cong_d^? y^{:T})$ $\Longrightarrow \exists \vec{z}:\ (\mathcal{U}\{x^{:S} \mapsto y^{:T}\} \wedge x^{:S} \cong_d^? y^{:T})$ if $x \in \mathcal{V}ar(\mathcal{U})$ and $y <_V x$
UMerge	$\exists \vec{z}:\ (\mathcal{U} \wedge x^{:T} \cong_d^? t^{:S} \wedge x^{:T'} \cong_d^? t'^{:S'})$ $\Longrightarrow \exists \vec{z}:\ (\mathcal{U} \wedge x^{:T\cup T'} \cong_d^? t^{:S} \wedge t^{:S} \cong_d^? t'^{:S'})$ if $x \in \mathcal{X}_\Diamond$ and $t^{:S} \notin \mathcal{X}_\Diamond$ and $\lvert t^{:S}\rvert \leq \lvert t'^{:S'}\rvert$
UErase	$\exists \vec{z}, z':\ (\mathcal{U} \wedge z'^{:S} \cong_d^? t^{:T})$ $\Longrightarrow \exists \vec{z}:\ (\mathcal{U})$ if $z' \notin \mathcal{V}ar(\mathcal{U}) \cup \mathcal{V}ar(t^{:T})$
UIntersect	$\exists \vec{z}:\ (\mathcal{U} \wedge (x :: \langle T\rangle)^{:S} \cong_d^? (y :: \langle T'\rangle)^{:S'})$ $\Longrightarrow \exists \vec{z}, z':\ \mathcal{U} \wedge x^{:S} \cong_d^? z'^{:\langle min(T\cup T')\rangle} \wedge y^{:S'} \cong_d^? z'^{:\langle min(T\cup T')\rangle}$ if $\langle T\rangle \bowtie_{\mathcal{S}_\Diamond}^{syn} \langle T'\rangle, z' :: \langle min(T \cup T')\rangle, z' \notin \mathcal{V}ar(\mathcal{U}) \cup \{x, y\}$

Figure 5: transformation rules of $\mathbf{UNIF_d}$

In the case of $\mathcal{T} = \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, σ may simply be called strict decorated unifier, $\mathcal{D}$-unifier, or $\mathcal{D}$-solution of $\mathcal{U}$ and the suffix $|_\mathcal{T}$ may be omitted. If $\mathcal{P}$ is clear from the context, we write $SU(\mathcal{U})$ instead of $SU_\mathcal{P}(\mathcal{U})$.

The notion of complete set of $\mathcal{T}$-unifiers is also relative to the subset $\mathcal{T}$:

Definition 6.8 *Let $\mathcal{U} = \bigwedge_{i\in I}(s_i \cong_d^? t_i)$ and $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ given in the specification $((\mathcal{S}, \mathcal{F}), \mathcal{P})$. $CSU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}}$ is a $\mathcal{T}$-complete set of unifiers of the unification problem $\mathcal{U}$, if:*

1. $CSU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}} \subseteq SU_\mathcal{P}(\mathcal{U})$, *(soundness)*
2. $\forall \phi \in SU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}}:\ (\exists \sigma \in CSU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}}:\ \sigma \lesssim_d^{\mathcal{V}ar(\mathcal{U})} \phi)$, *($\mathcal{T}$-completeness)*
3. $\forall \sigma \in CSU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}},\ \sigma \circ \sigma = \sigma$. *(idempotency)*

The complete set is minimal, *if any two different substitutions in $CSU_\mathcal{P}(\mathcal{U})_{|_\mathcal{T}}$ are incomparable with respect to $\lesssim_d^{\mathcal{V}ar(\mathcal{U})}$. If such a set is a singleton, this element is called* most general unifier *of $\mathcal{U}$.*

As before, we may omit $\mathcal{T}$ in the case $\mathcal{T} = \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ or $\mathcal{P}$ if the current presentation is non-ambiguous.

The unification process is described in two parts: the first in Figure 6.2 gives the transformation rules on non-degenerated problems, while the fail rules are given in Figure 6.2. Of course, the unification equation symbol $\cong_d^?$ is supposed to be commutative. An ordering $<_V$ on variables occurring in the problems is required for termination of the algorithm and extraction of solutions. For all $\mathcal{T} \subseteq \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, this set of rules is $\mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sound, $\mathcal{T}$-complete and terminating, as proved in [HK92], if the presentation is $\mathcal{T}$-sort inheriting w.r.t. $\leq_\mathcal{S}^{syn}$.

The essential difference between this approach and the classical one for regular unification with flat term declarations as in [JK91], is the treatment of unification of a variable with a non-variable

UConflict	$\exists\vec{x}:\ (\mathcal{U} \wedge f(t_1,\ldots,t_n)^{:S} \cong_d^? g(t'_1,\ldots,t'_p)^{:T})$ if $f \neq g$ or $S \not\approx T$	$\Longrightarrow \mathbb{F}$
UCheck*	$\exists\vec{x}:\ (\mathcal{U} \wedge x_1 \cong_d^? t_1[x_2]_{p_1} \wedge \ldots \wedge x_n \cong_d^? t_n[x_1]_{p_n})$ if $p_i \neq \wedge$ for some $i \in [1..n]$	$\Longrightarrow \mathbb{F}$
UDecoClash	$\exists\vec{x}:\ (\mathcal{U} \wedge x^{:S} \cong_d^? f(t_1,\ldots,t_n)^{:T})$ if $S \not\subseteq T$	$\Longrightarrow \mathbb{F}$
URemove	$\exists\vec{x}:\ (\mathcal{U} \wedge (x :: \langle T\rangle)^{:S} \cong_d^? (y :: \langle T'\rangle)^{:S'})$ if $\nexists\langle min(T \cup T')\rangle \in \mathcal{S}_\Diamond$	$\Longrightarrow \mathbb{F}$

Figure 6: fail rules of $\mathbf{UNIF_d}$

term. In the classical case, such an equation $x \cong_d^? t$ is solved by searching a substitution σ for the variables in t, such that the least sort of $\sigma(t)$ is lower than the sort of x, but still as general as possible (see the **Abstract** rule in [JK91]). This computation may be quite expensive. When flat term declarations are replaced by general ones and regularity is not assumed, it is even possible to prove the undecidability of the $(x \cong_d^? t)$-case (see [SS87]). However, special forms of term declarations, e.g. the so called semi-linear declarations, are still decidable [Uri92]. We think that these structural restrictions of term declarations are too rough, since there are surely many decidable theories that do not fulfill them. Instead, we only take the sorts in the decorations of potential images of variables into account (see **UDecoClash**). These decorations need to be constructed by a mechanism outside of the unification algorithm. It is easy to see that we do not change them in $\mathbf{UNIF_d}$. Hence, we pushed this source of undecidability outside of unification where more sophisticated mechanisms and properties of membership theories can be used.

Another source of undecidability is the unification of two variables in presentations with term declarations. The proof for the general unification can be found in [SS87] and for strict decorated unification in [Hin92]. The problem becomes trivially decidable by restricting to sort inheritance w.r.t. $\leq_{\mathcal{S}}^{syn}$. This property is also undecidable in general, but a sufficient condition is proposed in the sequel of this paper.

Proposition 6.9 *[HK92] The normal form obtained by applying the rules in* $\mathbf{UNIF_d}$ *to any unification problem is a solved form.*

A complete set of unifiers is derived by computing the tree-solved forms [JK91] of the results.

Proposition 6.10 *Let* $\mathcal{T} \subseteq ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $(\Sigma, \mathcal{P})$ *be a* $\mathcal{T}$*-sort inheriting specification w.r.t.* $\leq_{\mathcal{S}}^{syn}$. *Then the rules in* $\mathbf{UNIF_d}$ *are sound,* $\mathcal{T}$*-complete and terminate for every input unification problem* $\mathcal{U}$. *Let* $\mathcal{U}_{sf} = \bigwedge_{i\in I}(x_i :: A_i)^{:S_i} \cong_d^? t_i^{:T_i}$ *be the solved form obtained by the application of* $\mathbf{UNIF_d}$ *on a unification problem* $\mathcal{U}$ *and let* $\sigma = \bigcup_{i\in[1..n]}\sigma_i$ *with* $\sigma_i = \{t_i^{:T_i} \mapsto x_i^{:S_i}\}$ *if* $t_i \in \mathcal{X}_\Diamond$ *and* $x_i^{:S_i} <_V t_i^{:T_i}$ *and* $\sigma_i = \{x_i^{:S_i} \mapsto t_i^{:S_i}\}$ *otherwise. Then* $\{\sigma_{nf}\}$ *is a* $CSU_{\mathcal{P}}(\mathcal{U})_{|_{\mathcal{T}}}$ *where* σ_{nf} *is the tree-solved form of* σ.
If $\mathcal{U}_{sf} = \mathbb{F}$, *then* $\mathcal{U}$ *has no strict decorated* $\mathcal{T}$*-unifier and if* $\mathcal{U}_{sf} = \mathbb{T}$, *then any decorated substitution of terms*$(\mathcal{U})$ *into* $\mathcal{T}$, *is a strict decorated* $\mathcal{T}$*-unifier of* $\mathcal{U}$.

Proof: The proof can be found in [HK92]. It is worth noticing that the syntactic subsort relation is sufficient for the calculation of a $\mathcal{T}$-complete unifier of two variables, since $(\Sigma, \mathcal{P})$ is $\mathcal{T}$-sort inheriting and therefore there is no decoration of some subterm in $\mathcal{T}$ that contains incomparable sorts without common subsort. This fact allows us to subsume any subterm in $\mathcal{T}$ that may be an image of the two variables modulo sort inheritance. □

Corollary 6.11 *Let $\mathcal{T} \subseteq ValidT_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$ and $(\Sigma, \mathcal{P})$ be a $\mathcal{T}$-sort inheriting specification w.r.t. $\leq_S^{syn}$. Then, sound and $\mathcal{T}$-complete strict decorated unification is decidable.*

Finally, let us give two examples for the application of $\mathbf{UNIF_d}$:

Example 6.12 *Let $\mathcal{P} = \{x :: A, y :: B, x : B, a : A\}$ and:*

$$\mathcal{U} = (x^{:\{A\}} \cong_d^? a^{:\{A\}} \wedge x^{:\{A\}} \cong_d^? y^{:\{B\}} \wedge y^{:\{B\}} \cong_d^? a^{:\{A,B\}}).$$

This is transformed by **UCoalesce** *and* **UMerge** *into:*

$$\mathcal{U} = (x^{:\{A\}} \cong_d^? a^{:\{A\}} \wedge x^{:\{A\}} \cong_d^? y^{:\{B\}} \wedge a^{:\{A\}} \cong_d^? a^{:\{A,B\}}),$$

which clashes because of **UConflict**. *Hence, $\mathcal{U}$ has no $\mathcal{D}$-unifier, because $\sigma(x)$ should have the decorations $\{A\}$ and $\{A, B\}$ at the same time.*
Clearly, the required equality of the decorations of the potential images for x and y prohibited the solvability of $\mathcal{U}$, although $\mathcal{P} \models a : B$ is trivial in G-algebras. Only if the sort B is already in the decoration, we can unify them.

Example 6.13 *Let $\mathcal{P} = \{x :: A, y :: B, x : B, a : A\}$ and:*

$$\mathcal{U} = (x^{:\{A\}} \cong_d^? a^{:\{A,B\}} \wedge x^{:\{A\}} \cong_d^? y^{:\{B\}} \wedge y^{:\{B\}} \cong_d^? a^{:\{A,B\}}).$$

This is transformed by **UCoalesce**, **UMerge** *and* **Delete** *yielding:*

$$\mathcal{U} = (x^{:\{A\}} \cong_d^? a^{:\{A,B\}} \wedge x^{:\{A\}} \cong_d^? y^{:\{B\}}),$$

which is in solved form. Hence, $\mathcal{U}$ has the $\mathcal{D}$-unifier:

$$\sigma = \{x^{:\{A,B\}} \mapsto a^{:\{A,B\}}, y^{:\{A,B\}} \mapsto a^{:\{A,B\}}\}.$$

6.3 Subterm Conservative Solutions

In order to solve typing problems, we need a notion describing that in solutions consisting of substitution sets the images of the substitutions are subterms of the original problem. This gives us the bottom-up typability of the terms in the image, if the terms in the initial problem were bottom-up typable – a very useful property for doing some of the induction proofs needed in the sequel.

Definition 6.14 *Let $\mathcal{C}$ be a class of problems Pb, whose minimal, complete set of solutions can be described as a set of decorated substitutions $MCSol(Pb)$ in dag-solved form (see [JK91]).*

$\mathcal{C}$ is called subterm conservative *if for all problems $Pb \in \mathcal{C}$ there is a subterm conservative $MCSol(Pb)$, i.e. for all $\sigma \in MCSol(Pb)$ and for all terms $t \in \mathcal{I}m(\sigma)$ there exist a decorated substitution τ with $\mathcal{R}an(\tau) = \mathcal{I}m(\tau)$, s.t. $\forall z^{:S} \in \mathcal{R}an(\tau)$ with $z :: C : S \approx \{C\}$, and a $t' \in subterm\_set(Pb)$* s.t. $t \cong_d \tau(t')$.

$\mathcal{C}$ is called strictly subterm conservative *if $\mathcal{C}$ is subterm conservative and $\mathcal{D}om(\tau) = \emptyset$, i.e. τ is the identity substitution.*

Therefore a problem class is subterm conservative, if there is a way to express a minimal complete set of solutions by a set of dag-solved form substitutions constructed from subterms in the original problem modulo variable renaming/specialization (a notion from [Wal92]), resp. strictly subterm conservative if there is no variable renaming.

Proposition 6.15 *Let $(\Sigma, \mathcal{P})$ be a sort inheriting specification. Then strict decorated unification is subterm conservative and strict decorated matching is strictly subterm conservative.*

Proof: The following proof is only a sketch.

For strict decorated matching, the claim is trivial since we can only change the variables of the left-hand side of a matching equation. Therefore the introduction of a subterm that does not belong to the original problem leads immediately to a contradiction with the definition of a solution.

In the case of strict decorated unification, we can use the fact that $\mathbf{UNIF_d}$ yields a complete solution set as singleton $\{\sigma\}$. Therefore the minimal complete set of solutions is unique up to variable renaming.

No rule in the unification algorithm decomposes a term that is bound to a variable, neither constructs a new term. The only new term parts introduced are variables. But this is done by decreasing the sort w.r.t. the 'old' variable which the new one is bound to. The decoration S of such new variables $z :: C$ clearly satisfy $S \approx \{C\}$ by construction.

Together with the fact that variable decorations of new variables do never change, since there are by definition no old occurrences of the same variable, this guarantees that for any substitution σ in the solution set, there must also be a decorated substitution τ with $\mathcal{R}an(\tau) = \mathcal{I}m(\tau)$ and a $t' \in subterm\_set(Pb)$ s.t. $t \cong_d \tau(t')$ for all terms in $\mathcal{I}m(\sigma)$. □

This proposition would not necessarily hold if we were using term declarations during the unification, like for unification in non-regular theories, or other mechanisms introducing new term parts, like equational theories, etc. However, semantical arguments could help us finding the typability of these newly introduced term parts.

7 Decorated Term Rewriting Systems

We introduce in this section the two kinds of rewrite rules that provide the basis of the deduction process.

7.1 Decorated Equalities

Definition 7.1 *A* decorated equality *is a pair of two decorated terms, denoted by* $(p^{:S} = q^{:S'})$ *where* $p^{:S}, q^{:S'} \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$.

The next definition of replacement of equal be equal, as well as the definition of a rewriting step, are given for a set of occurrences in order to later define a notion of parallel reduction.

Definition 7.2 *A decorated term* $t^{:S}$ is equal *to* $t'^{:S'}$, *using the set of decorated equalities* E, *if there exist a set of incomparable (w.r.t.* $\preceq$*) occurrences* $O \neq \emptyset$ *with* $t^{:S} \cong_d t^{:S}[u^{:U}]_O$, $t'^{:S'} \cong_d t'^{:S'}[u'^{:U'}]_O$, *a decorated substitution* σ *and a decorated equality* $\phi = (p^{:S} = q^{:S'})$ *in* E, *such that* σ *is a solution of* $(p^{:S} \lesssim^?_d u^{:U} \wedge q^{:S'} \lesssim^?_d u'^{:U'})$. *This is denoted by* $t^{:S} \longleftrightarrow_E^{O,\sigma,\phi} t'^{:S'}$. *If* O *is a singleton* $\{\omega\}$, *we write* $t^{:S} \longleftrightarrow_E^{\omega,\sigma,\phi} t'^{:S'}$

7.2 Decorated Rewriting

A decorated rewrite rule is just a pair of decorated terms. It is applied on a decorated term by $\mathcal{D}$-matching the left-hand side to some subterm.

Definition 7.3 *A* decorated rewrite rule *is a pair of decorated terms denoted by* $(l^{:S_l} \to r^{:S_r})$, *where* $l^{:S_l}, r^{:S_r} \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, *such that* $S_l \subsetneq S_r$ *and* $\mathcal{V}ar(l^{:S_l}) \supseteq \mathcal{V}ar(r^{:S_r})$. *A* decorated rewrite system *is a set of decorated rewrite rules.*

Definition 7.4 *A decorated term* $t^{:S}$ rewrites *to* $t'^{:S'}$ *using the decorated rewrite system* R *if there exist an occurrence set* $O \neq \emptyset$ *with* $t^{:S} \cong_d t^{:S}[u^{:U}]_O$, *a decorated substitution* σ *and a decorated rewrite rule* $\phi = (l^{:S_l} \to r^{:S_r})$ *in* R *such that:*

1. σ *is a* $\mathcal{D}$*-match from* $l^{:S_l}$ *to* $u^{:U}$,
2. $t'^{:S'} =_d t^{:S}[\sigma(r^{:S_r})]_O$.

This is denoted by $t^{:S} \rightarrowtail_R^{O,\sigma,\phi} t'^{:S'}$. *If* O *is a singleton* $\{\omega\}$, *we also write* $t^{:S} \rightarrowtail_R^{\omega,\sigma,\phi} t'^{:S'}$. *The symmetric closure of* $\rightarrowtail_R$ *is written* $\leftarrow\!\!\diamond\!\!\rightarrowtail_R$.

Notice the accumulation of decorations at the redex occurrence, due to the condition $S_l \subsetneq S_r$ for decorated rewrite rules.

Example 7.5 *Let* $a^{:\{A\}} \to b^{:\{A,B\}}$ *be a decorated rewrite rule. Then* $f^{:\emptyset}(a^{:\emptyset})$ *cannot be rewritten, but* $f^{:\emptyset}(a^{:\{A\}}) \rightarrowtail f^{:\emptyset}(b^{:\{A,B\}})$.

The symmetric closure of the rewrite relation is written $t^{:S} \leftarrow\!\!\diamond\!\!\rightarrowtail_R^{\omega,\sigma,\phi} t'^{:S'}$. We can conclude, that $\rightarrowtail_E$ is a conservative extension of $\rightarrowtail_R$, i.e. any decorated rewriting step can be seen as application of an equality.

7.3 Decoration Rewriting

Then we need to introduce decoration rewrite rules, whose purpose is to locally increment the set of sorts associated to some node in a decorated term. In order to formalize this process as a reduction process, a new set of variables is introduced. Let us denote the elements of $\mathbf{V}$ by s, s', s''. A decoration rewrite rule is then given by a decorated term, say l, a set-variable $s \in \mathbf{V}$ that decorates the left-hand side, a sort expression $s \cup S_l$ where S_l is a set of sorts to decorate the right-hand side, and a condition $c(s)$ depending on the variable s. Applying to $t^{:U}$ a decoration rewrite rule $(l^{:s} \to l^{:s \cup S_l}$ if $c(s))$ amounts to $\mathcal{D}$-match the decorated term $l^{:U}$ to the decorated term $t^{:U}$, to check the condition $c(s)$ when s takes the value U and to enrich the decoration U with S_l if $c(U)$ is true.

Definition 7.6 *A* decoration rewrite rule *is a conditional rewrite rule denoted by* $(l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s)$ *where* $l^{:S_l} \in \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$, *such that all variables with multiple occurrences have unique decorations,* $s \in \mathbf{V}$ *and* S_l *is a set of sorts. A* decoration rewrite system *is a set of decoration rewrite rules.*

Definition 7.7 *A decorated term* $t^{:S}$ rewrites *to* $t'^{:S'}$ *using the decoration rewrite system* D *if there exist a set of occurrences* $O \neq \emptyset$ *with* $t^{:S} \cong_d t^{:S}[u^{:U}]_O$, *a decorated substitution* σ *and a decoration rewrite rule* $\phi = (l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s)$ *such that:*

1. σ *is a* $\mathcal{D}$*-match from* $l^{:U}$ *to* $u^{:U}$,
2. $S_l \not\subseteq U$,
3. $t'^{:S'} =_d t^{:S}[\sigma(l^{:U})^{:U \cup S_l}]_O$.

This is denoted by $t^{:S} \rightarrowtail_D^{O,\sigma,\phi} t'^{:S'}$. *If* O *is a singleton* $\{\omega\}$, *this is also written* $t^{:S} \rightarrowtail_D^{\omega,\sigma,\phi} t'^{:S'}$. *The symmetric closure of* $\rightarrowtail_D$ *is written* $\leftarrow\!\!\diamond\!\!\rightarrowtail_D$.

Example 7.8 *Let* $(a^{:s} \to a^{:s \cup \{A\}}$ if $\{A\} \not\subseteq s)$ *be a decoration rewrite rule. Then* $a^{:\{B\}} \rightarrowtail_D a^{:\{A,B\}}$. *Let* $(x^{:s} \to x^{:s \cup \{A,B,C\}})$ if $\{A,B,C\} \not\subseteq s$ *be another decoration rewrite rule. Therefore* $a^{:\{A,B\}} \rightarrowtail_D a^{:\{A,B,C\}}$.

Given a pair (D, R) of decorated rewrite rules and decoration rules, we define for two decorated terms $t^{:S}$ and $t'^{:S'}$, $t^{:S} \rightarrowtail_{D \cup R} t'^{:S'}$ if $t^{:S} \rightarrowtail_R t'^{:S'}$ or $t^{:S} \rightarrowtail_D t'^{:S'}$.

From now on, $rhs(\phi)$ and $lhs(\phi)$ denote as usual respectively the right-hand side and left-hand side of a decorated/decoration rewrite rule ϕ.

7.4 Elementary Properties of Rewriting

Let us first introduce the notions of canonical validity and theorems associated to sets of decorated and decoration rewrite rules and equalities.

Definition 7.9 *For any sets of decoration rewrite rules D, decorated rewrite rules R and decorated equalities E, we say that:*

1. $(l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s) \in D$ *is* valid *in* $\mathcal{P}$, *if* $l^{:S_l}$ *is valid in* $\mathcal{P}$.
2. $(l^{:S_l} \to r^{:S_r}) \in R$ *is* valid *in* $\mathcal{P}$, *if* $l^{:S_l}, r^{:S_r}$ *are valid in* $\mathcal{P}$ *and for all common* $\mathcal{T}(\Sigma, \mathcal{X})$*-assignments* α *of* $l^{:S_l}, r^{:S_r}$, $\mathcal{P} \vdash \alpha((l^{:S_l})_{nd}) = \alpha((r^{:S_r})_{nd})$.
3. $(p^{:S_p} = q^{:S_q}) \in E$ *is* valid *in* $\mathcal{P}$ *if* $p^{:S_p}, q^{:S_q}$ *are valid in* $\mathcal{P}$ *and for all common* $\mathcal{T}(\Sigma, \mathcal{X})$*-assignments* α *of* $p^{:S_p}, q^{:S_q}$, $\mathcal{P} \vdash \alpha((p^{:S_p})_{nd}) = \alpha((q^{:S_q})_{nd})$.

Definition 7.10 *To any sets of decoration rewrite rules D, of decorated equalities E and decorated rewrite rules R, we associate canonically a set of formulas $Th(\mathcal{P}_{DER})$, that is the union of:*

1. $\{t = t' \mid t, t' \in \mathcal{T}(\Sigma, \mathcal{X})$ and $\exists t_0 \in \mathcal{T}_d(\mathcal{S}_\emptyset, \mathcal{F}, \mathcal{X}_\emptyset), A \in \mathcal{S}_\emptyset$, s.t. $t^{:\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t_0^{:\{A\} \cup U} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:\emptyset}\}$,
2. $\{t : A \mid t \in \mathcal{T}(\Sigma, \mathcal{X})$ and $\exists t' \in \mathcal{T}_d(\mathcal{S}_\emptyset, \mathcal{F}, \mathcal{X}_\emptyset), A' \in \mathcal{S}_\emptyset$ with $A' \leq^{syn}_{\mathcal{S}_\emptyset} \langle A \rangle$, s.t. $t^{:\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:\{A'\} \cup U}\}$ *and*
3. $\{EX\ t \mid t \in \mathcal{T}(\Sigma, \mathcal{X})$ and $\exists t' \in \mathcal{T}_d(\mathcal{S}_\emptyset, \mathcal{F}, \mathcal{X}_\emptyset), A \in \mathcal{S}_\emptyset$, s.t. $t^{:\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:\{A\} \cup U}\}$.

The following properties of rewriting are easy to check but quite useful in what follows.

Lemma 7.11 *Let $t^{:S}$ be a valid term and ϕ a valid element of $D \cup E \cup R$ in $\mathcal{P}$. If $t^{:S} \overset{\omega,\sigma,\phi}{\longleftrightarrow}_{D \cup E \cup R} t'^{:S'}$ then:*

1. $\forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'})$: $(\mathcal{P} \vdash \alpha((t^{:S})_{nd}) : A)$ *iff* $(\mathcal{P} \vdash \alpha((t'^{:S'})_{nd}) : A)$,
2. $\forall \langle A, \ldots \rangle \in S' \setminus S$, $\exists \alpha' \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'})$: $(\mathcal{P} \vdash \alpha'((t^{:S})_{nd}) : A)$,
3. $\forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'})$: $(\mathcal{P} \vdash EX\ \alpha((t'^{:S'})_{nd})) \Rightarrow (\mathcal{P} \vdash \alpha((t^{:S})_{nd}) = \alpha((t'^{:S'})_{nd}))$ *and*
4. $t'^{:S'}$ *is also valid in* $\mathcal{P}$,

Proof: (1) If $\phi \in D$, then $t^{:S} =_{nd} t'^{:S'}$ and therefore the property is trivial. Otherwise $\phi \in E \cup R$ and we can easily construct a proof using the axioms **Symmetry**, **EqSubstitutivity** and **MeReplacement**, where the validity of the terms in $\mathcal{I}m(\sigma)$ is needed. The validity of these terms is automatically provided if $\phi \in R$ and the rule is applied left-to-right, since $t^{:S}$ is valid and strict decorated matching is subterm conservative.

(2) This follows immediately from (4) and (1).

(3) With the help of **Reflexivity**, **Symmetry**, **EqSubstitutivity** and **EqReplacement**, we can easily construct the needed proof.

(4) If $\phi \in D$, we can use the formula corresponding to ϕ and apply **MeSubstitutivity** to it, when ϕ is applied left-to-right, else the validity of $t'^{:S'}$ is subsumed by the one of $t^{:S}$.

Otherwise $\phi \in E \cup R$ and we have to take the different cases for the occurrences ν into consideration, where we want to prove the validity of the decoration. If $\nu \asymp \omega$, then validity of $(t'^{:S'})_{|_\nu}$ is equivalent to the one of $(t^{:S})_{|_\nu}$. If $\nu \preceq \omega$, then we can construct a proof as in (1). If $\nu \succ \omega$, then the validity of $(t'^{:S'})_{|_\nu}$ follows from the validity of ϕ and those in $\mathcal{I}m(\sigma)$, using axiom **MeSubstitutivity**. Remark that the validity of the terms in $\mathcal{I}m(\sigma)$ is subsumed by the one of $t^{:S}$ if $\phi \in R$ is applied left-to-right. □

Analyzing the proof of lemma 7.11 for the left-to-right application of ϕ leads to:

Corollary 7.12 *Let $t^{:S}$ and ϕ be valid in $\mathcal{P}$. If $t^{:S} \rightarrowtail^{\omega,\sigma,\phi}_{D \cup R} t'^{:S'}$ then all terms in $\mathcal{I}m(\sigma)$ are valid. Furthermore:*

1. $(\exists \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S}) : (\mathcal{P} \vdash \alpha((t^{:S})_{nd}) : A))$ *iff*
$(\exists \alpha' \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'}) : (\mathcal{P} \vdash \alpha'((t'^{:S'})_{nd}) : A))$,

2. $\forall \langle A, \ldots \rangle \in S' \setminus S,\ \exists \alpha' \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'}) :$
$(\mathcal{P} \vdash \alpha'((t^{:S})_{nd}) : A)$,

3. $\forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'}) :$
$(\mathcal{P} \vdash EX\ \alpha((t'^{:S'})_{nd})) \Rightarrow (\mathcal{P} \vdash \alpha((t^{:S})_{nd}) = \alpha((t'^{:S'})_{nd}))$ *and*

4. $t'^{:S'}$ *is also valid in* $\mathcal{P}$,

Therefore we don't need to check the $\mathcal{P}$-validity in rewriting proofs since clearly, this result extends to multiple rewrite rule applications.

The following statement tells that a decoration can be inherited along replacement of equal by equal using valid D, E and R.

Lemma 7.13 *For all decorated terms $t^{:S}, t'^{:S'}$, where $t^{:S}$ is valid and $t^{:S} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:S'}$,*

$$\forall \langle A, \ldots \rangle \in S \cup S', \forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:S}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'}),\ (\mathcal{P} \vdash \alpha((t^{:S})_{nd}) : A).$$

Thus if $t^{:S} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:S'}$, then $t^{:S \cup S'}$ is a valid decorated term.

Proof: If A belongs to S, it is obvious by the validity of $t^{:S}$. Otherwise if A belongs to $S' \setminus S$, let us prove it by induction on the length of $\overset{*}{\longleftrightarrow}_{D \cup E \cup R}$. If this length is zero the result is clear. Otherwise

$$t^{:S} \longleftrightarrow_{D \cup E \cup R} t_1{}^{:S_1} \overset{*}{\longleftrightarrow}_{D \cup E \cup R} t'^{:S'}$$

and by the induction hypothesis, $\langle A, \ldots \rangle \in S'$ implies that there is an extension α' of α, s.t. $\mathcal{P} \vdash \alpha'((t_1{}^{:S_1})_{nd}) : A$. Now $(\alpha((t^{:S})_{nd}) : A)$ is a consequence of Lemma 7.11. □

This usefully extends to empty decorated terms:

Lemma 7.14 *If $t^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{D \cup R} t'^{:S'}$, then:*

1. $\forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:\downarrow\emptyset}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'})$:
$(\mathcal{P} \vdash \alpha((t^{:\downarrow\emptyset})_{nd}) : A)$ *iff* $(\mathcal{P} \vdash \alpha((t'^{:S'})_{nd}) : A)$,

2. $S' \neq \emptyset \Rightarrow \forall \alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:\downarrow\emptyset}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:S'})$:
$(\mathcal{P} \vdash \alpha((t^{:\downarrow\emptyset})_{nd}) = \alpha((t'^{:S'})_{nd}))$.

3. $t'^{:S'}$ *is valid in* $\mathcal{P}$,

Proof: This can easily be shown by induction on the length of the rewriting proof. If the length is zero, $t^{:\downarrow\emptyset} \cong_d t'^{:S'}$ and therefore the lemma is trivial. Otherwise we cut off the last step:

$$t^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{D \cup R} t_0^{:S_0} \rightarrowtail_{D \cup R} t'^{:S'}.$$

The induction hypothesis gives us the validity of $t_0^{:S_0}$ and the equivalence of membership for $t^{:\downarrow\emptyset}{}_{nd}$ and $(t_0^{:S_0})_{nd}$. For the last step we now can apply corollary 7.12. □

In order to show that the validity-check is needed in the other direction, consider the following example:

Example 7.15 *Let* $D \cup R$ *be:*

$$\begin{array}{rcll} f(x^{:\{A\}}, y^{:\{B\}})^{:\emptyset} & \rightarrow & x^{:\{A\}} & \\ x^{:s} & \rightarrow & x^{:s \cup \{A\}} & \text{if } \{A\} \not\subseteq s \\ y^{:s} & \rightarrow & y^{:s \cup \{B\}} & \text{if } \{B\} \not\subseteq s \\ a^{:s} & \rightarrow & a^{:s \cup \{A\}} & \text{if } \{A\} \not\subseteq s \end{array}$$

Then without checking the validity of the terms in the image of the used substitution we would have $a^{:\{A\}} \overset{*}{\leftrightarrow}_R f(a^{:\{A\}}, b^{:\{B\}})^{:\emptyset}$, *but* $f(a^{:\{A\}}, b^{:\{B\}})^{:\emptyset}$ *is not valid, because of* $b^{:\{B\}}$, *since* b *does not belong to* B *in* $D \cup R$.

Clearly, if we allowed non-valid terms in the range of substitutions, then we could prove arbitrary membership formulas, provided we interpret all terms generated by $\overset{*}{\leftrightarrow}$ as valid.

Another consequence of Lemma 7.11 is the soundness of proofs in $\overset{*}{\leftrightarrow}_{D \cup E \cup R}$.

Lemma 7.16 *Let* D *be a set of decoration rewrite rules,* R *a set of decorated rewrite rules and* E *a set of decorated equations, s.t. all* $\phi \in D \cup E \cup R$ *are valid.*
Then $Th(\mathcal{P}_{DER}) \subseteq Th(\mathcal{P})$

Proof: Let ϕ be a formula in $Th(\mathcal{P}_{DER})$. Then Definition 7.10 says, that there must be a proof

$$\begin{array}{ll} t^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D \cup E \cup R} t_0^{:\{A'\} \cup U} \overset{*}{\leftrightarrow}_{D \cup E \cup R} t'^{:\downarrow\emptyset} & \text{if } \phi = (t = t'), \\ t^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D \cup E \cup R} t_0^{:\{A'\} \cup U} & \text{if } \phi = (t : A) \text{ and} \\ t^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D \cup E \cup R} t_0^{:\{A'\} \cup U} & \text{if } \phi = (EX\ t), \end{array}$$

s.t. $A' \leq_{S_0}^{syn} \langle A \rangle$, if $\phi = (t : A)$. Let w.l.o.g. $A' = \langle B, \ldots \rangle$ with $B \leq_S^{syn} A$ in this case.

Since $\overset{*}{\leftrightarrow}_{D \cup E \cup R}$ preserves membership and validity of terms (Lemma 7.13) and $t^{:\downarrow\emptyset}$ is valid, we are sure that for all $\alpha \in \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t_0^{:\{A'\} \cup U}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:\downarrow\emptyset})(\cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:\downarrow\emptyset})$ if $\phi = (t = t'))$, $\alpha((t_0^{:\{A'\} \cup U})_{nd})$ belongs to B in $\mathcal{P}$, as well as $\alpha((t^{:\downarrow\emptyset})_{nd})$ (and $\alpha((t'^{:\downarrow\emptyset})_{nd})$). Hence, **MeSubstitutivity** implies that they also belong to A, if $\phi = (t : A)$. Furthermore, the same Lemma ensures $\mathcal{P} \vdash \alpha((t^{:\downarrow\emptyset})_{nd}) = \alpha((t_0^{:\{A'\} \cup U})_{nd})$ (and $\mathcal{P} \vdash \alpha((t'^{:\downarrow\emptyset})_{nd}) = \alpha((t_0^{:\{A'\} \cup U})_{nd})$ if $\phi = (t = t')$. Consequently, $\mathcal{P} \vdash \phi$, since t (and t') are in $\mathcal{T}(\Sigma, \mathcal{X})$. Therefore an α with $\alpha_{|_{Var(t)}} = id$ must exist in $\mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t_0^{:\{A'\} \cup U}) \cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t^{:\downarrow\emptyset})(\cap \mathcal{ASS}_{\mathcal{T}(\Sigma,\mathcal{X})}(t'^{:\downarrow\emptyset})$ if $\phi = (t = t'))$. Note that we are sure that all sorts are non-empty, thanks to assumption 4.6. □

Decorated and decoration rewriting are stable by context and substitution:

Proposition 7.17 *If* $t^{:S} \rightarrowtail^{\omega,\sigma,\phi}_{D\cup R} t'^{:S'}$:

1. *for any decorated substitution* σ *for* $t^{:S}$, $\sigma(t^{:S}) \rightarrowtail^{\omega,\sigma\sigma,\phi}_{D\cup R} \sigma(t'^{:S'})$,
2. *for any decorated term* u *and any position* v *in* u, $u[t^{:S}]_v \rightarrowtail^{\omega,\sigma,\phi}_{D\cup R} u[t'^{:S'}]_v$.

Proof: This comes from the fact that substitution and context act on both sides of the rewrite step in an identical way, thus leaving the sort information in the same respective situation. Let us detail this. Assume that $t^{:S} \rightarrowtail^{\omega,\sigma,\phi}_{D\cup R} t'^{:S'}$ and $\phi : l^{:S_l} \rightarrow r^{:S_r}$ respectively $\phi : l^{:s} \rightarrow l^{:s\cup T}$ if $T \not\subseteq s$. Let furthermore $t^{:S} =_d t^{:S}[u^{:U}]_\omega$.
In the case of decorated rewriting, $\sigma(l^{:S_l}) \cong_d t^{:S}{}_{|\omega}$ and thus for the context $C[]$ whose hole is at occurrence v we have $\sigma(l^{:S_l}) \cong_d C[t^{:S}]_{|v.\omega}$ which proves that $C[t]$ rewrites to $C[t']$.
In the case of decoration rewriting, $\sigma(l^{:U}) =_d (t_{|\omega})^{:U}$ and thus for the context $C[]$ whose hole is at occurrence v we have $\sigma(l^{:U}) \cong_d C[t^{:S}]_{|v.\omega}$ which proves that $C[t]$ rewrites to $C[t']$.
The stability by substitution is a consequence of the definitions. □

As a consequence, the symmetric relation $\leftrightarrowtail$ is also stable by context and substitution.

A last property allows us to work with representatives of $\cong_d$-equivalence classes when proving rewriting proof properties in the sequel.

Proposition 7.18 *Let* $t, t', \bar{t}, \bar{t}' \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ *s.t.* $t \cong_d t'$. *Then* $t \rightarrowtail^{\omega,\sigma,\phi}_{D\cup R} \bar{t}$ *and* $t' \rightarrowtail^{\omega,\sigma',\phi}_{D\cup R} \bar{t}'$ *implies* $\bar{t} \cong_d \bar{t}'$.

Proof: Let $t_{|\omega} =_d u^{:U}$ and $\bar{t}_{|\omega} =_d u'^{:U'}$ and $\phi \in R$ be $l^{:S_l} \rightarrow r^{:S_r}$ (resp. $\phi \in D$ and $l^{:s} \rightarrow l^{:s\cup S_l}$ if $S_l \not\subseteq s$). Clearly, $t[\sigma(l^{:S_l})] \cong_d t \cong_d t' \cong_d t'[\sigma'(l^{:S_l})]$ (resp. $t[\sigma(l^{:U})] \cong_d t \cong_d t' \cong_d t'[\sigma'(l^{:U})]$). Therefore $\sigma \cong_d^{\mathcal{V}ar_d(t)} \sigma'$ and consequently $t[\sigma(r^{:S_r})] \cong_d t'[\sigma'(r^{:S_r})]$ (resp. $t[\sigma(l^{:U})^{:U\cup S_l}] \cong_d t'[\sigma'(l^{:U})^{:U\cup S_l}]$).
□

7.5 Converting Presentations to Decorated TRSs

Given a presentation $\mathcal{P}$, we describe in Figure 7 how to extract a set of decorated equalities $E_\mathcal{P}$ and a set of decoration rules $D_\mathcal{P}$. Remark that all variables in the constructed rules are supposed to have their sort in their decoration – otherwise any rule ϕ containing a variable in $t = lhs(\phi)$ would not be applicable on t itself, since the needed substitution would not fulfill the conditions for decorated substitutions.

The first kind of rules allows using existential declarations, the second term declarations. It is superfluous to introduce decoration rules corresponding to variable declarations $(x :: A)$, thanks to the matching modulo sort inheritance. The sort Ω and the first kind of rules are useful at a theoretical level to deduce existence formulas. In the sequel, we most often omit Ω on the decorations in order to ease notations.

The application of the decoration rules corresponds with a partial typing process, adding all syntactic sorts of a term to its decorations.

Example 7.19 *Let us consider the following specification* $\mathcal{S}_1$ *which set of sorts is* $\{A, B\}$ *the set of operators is* $\{f, +, a, b\}$ *and satisfying the following formulas:*

1. $(u^{:\downarrow\emptyset})^{:s} \rightarrow (u^{:\downarrow\emptyset})^{:s\cup\{\Omega\}}$ if $\{\Omega\} \not\subseteq s$
   For each subterm u of a term v appearing in a formula in $\mathcal{P}$,
   For such a rule, $u^{:\downarrow\emptyset} \in \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$, $s \in \mathbf{V}$.

2. $(u^{:\downarrow\emptyset})^{:s} \rightarrow (u^{:\downarrow\emptyset})^{:s\cup\{A\}}$ if $\{A\} \not\subseteq s$
   For all $(u : A)$ in $\mathcal{P}$.
   For such a rule, $u^{:\downarrow\emptyset} \in \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$, $s \in \mathbf{V}$, $A \in \mathcal{S}$.

$D_\mathcal{P}$: decoration rules associated to $\mathcal{P}$

1. $p^{:\downarrow\emptyset} = q^{:\downarrow\emptyset}$
   For each equality $(p = q)$ in $\mathcal{P}$

$E_\mathcal{P}$: decorated equalities associated to $\mathcal{P}$

Figure 7: Extraction of decorated/decoration rewrite rules from the Presentation

$$
\begin{array}{rcll}
x & :: & A & (1)\\
y & :: & B & (2)\\
z & :: & B & (3)\\
x & : & B & (4)\\
x+b & : & A & (5)\\
 & & & \\
f(x) & : & A & (6)\\
f(y) & : & B & (7)\\
y+z & : & B & (8)\\
a & : & A & (9)\\
b & : & B & (10)\\
 & & & \\
f(x) & = & x & (11)\\
a+y & = & b. & (12)
\end{array}
$$

Declarations 1 and 4 correspond to specifying the subsort declaration $A \leq_\mathcal{S}^{syn} B$. *From this specification, the following set of decorated rewrite rules* D_1 *is generated:*

$$
\begin{array}{llll}
x^{:s} \rightarrowtail x^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s & f(x^{:\{A\}})^{:s} \rightarrowtail f(x^{:\{A\}})^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s\\
a^{:s} \rightarrowtail a^{:s\cup\{A\}} & \text{if } A \notin s & f(y^{:\{B\}})^{:s} \rightarrowtail f(y^{:\{B\}})^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s\\
b^{:s} \rightarrowtail b^{:s\cup\{B\}} & \text{if } B \notin s & (x^{:\{A\}} + b^{:\emptyset})^{:s} \rightarrowtail (x^{:\{A\}} + b^{:\emptyset})^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s\\
 & & (y^{:\{B\}} + z^{:\{B\}})^{:s} \rightarrowtail (y^{:\{B\}} + z^{:\{B\}})^{:s\cup\{B\}} & \text{if } B \notin s.
\end{array}
$$

We have omitted the decoration rules involving the sort Ω *that might irritate by their number and are not really useful in order to see the principles of decoration rule application, since they are treated in the same way. The term* $a^{:\emptyset} + b^{:\emptyset}$ *can be decorated, using* D_1 *in the following way:*

$$(a^{:\emptyset} + b^{:\emptyset})^{:\emptyset} \rightarrowtail_{D_1} (a^{:\{A\}} + b^{:\emptyset})^{:\emptyset} \rightarrowtail_{D_1} (a^{:\{A\}} + b^{:\emptyset})^{:\{A\}} \rightarrowtail_{D_1} (a^{:\{A\}} + b^{:\{B\}})^{:\{A\}}$$

Notice that 11 and 12 are not yet used nor yet transformed at this point.

This rewrite based decoration process is a restricted application of the deduction rules of G-algebra concerned only by membership formula deduction. The decoration process does not use the equational part of the specification for deriving new sorts from equalities. It is thus in general incomplete in the sense that it does not compute all the possible sorts of a term, even when this is computable.

7.6 Decorated Orderings

Finally, in order to find an extension of the classical reduction ordering over undecorated terms, we compare the decorations of $=_{nd}$-equal terms:

Definition 7.20 *Let $t^{:S}$ and $t'^{:S'}$ be decorated terms, such that $t^{:S} =_{nd} t'^{:S'}$. $t^{:S}$ is said* less decorated *than $t'^{:S'}$, written $t^{:S} \Subset t'^{:S'}$ if $\forall \omega \in \mathcal{D}om(t), Deco(t_{|\omega}) \subsetneq_{\mathcal{P}} Deco(t'_{|\omega})$. The relation $\Subset$ is called decoration subsumption ordering.*

The classical notion of reduction ordering can now be extended on decorated terms in a straightforward way. Such an ordering is useful for termination proofs.

Definition 7.21 *Let $((\mathcal{S}, \mathcal{F}), \mathcal{P})$ be a presentation with bounded membership. A quasi-ordering $>_d$ over $\mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$ is a* decorated reduction ordering *w.r.t. to a rule set R, if there is an (undecorated) reduction ordering $>$ on $\mathcal{T}(\Sigma, \mathcal{X})$, s.t.:*

1. *$t >_d t'$* iff *$(t_{nd}, t)(>, \Subset)(t'_{nd}, t')$,*
   i.e. $>_d$ is the lexicographic ordering based on the undecorated ordering and the decoration subsumption over pairs (t_{nd}, t),
2. *$>_d$ is compatible with R, i.e. $\forall (l \to r) \in R$, $l >_d r$.*

A last notion, necessary for later induction proofs, is the downward completeness of a set of decorated terms w.r.t. a given ordering.

Definition 7.22 *Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F})$ and $>_d$ be a quasi-ordering over $\mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F})$. $\mathcal{T}$ is* downward complete *iff for all $t^{:S}, t'^{:S'}$ the following holds:*

$$t^{:S} \in \mathcal{T} \text{ and } t^{:S} >_d t'^{:S'} \Rightarrow t'^{:S'} \in \mathcal{T}.$$

8 A Birkhoff Theorem for G-Algebra

In order to get an executable version of deduction in G-algebra, we need to prove a Birkhoff-like theorem stating that two terms can be proved to be equivalent using the deduction rules of G-algebra iff they can be proved equivalent by replacement of equal by equal on decorated terms.

Given a set E of decorated equalities and a set D of decoration rewrite rules, extracted for instance from a presentation $\mathcal{P}$ as in the previous section, $\longleftrightarrow\!\!\diamond\!\!\rightarrow_{D \cup E}$ is $\longleftrightarrow\!\!\diamond\!\!\rightarrow_E \cup \longleftrightarrow\!\!\diamond\!\!\rightarrow_D$ and $\overset{*}{\longleftrightarrow\!\!\diamond\!\!\rightarrow}_{D \cup E}$ its reflexive, symmetric and transitive closure.

Theorem 8.1 *Let t, t' be two terms in $\mathcal{T}(\Sigma, \mathcal{X})$, $\mathcal{P}$ be a Σ-presentation. Let furthermore $D_\mathcal{P}$ and $E_\mathcal{P}$ be resp. sets of decoration rewrite rules and equational axioms associated to $\mathcal{P}$ and $R = \emptyset$. Then $Th(\mathcal{P}) = Th(\mathcal{P}_{D_\mathcal{P} E_\mathcal{P} R})$.*

The proof is rather technical and long. It relies on the step by step description of the correspondence between deductions and equational proofs on decorated terms and can be skipped in a first reading of the paper.

The first step consists in proving that the use of $\longleftrightarrow\!\!\diamond\!\!\rightarrow_{D \cup E}$ is sound with respect to G-deduction. This is an obvious consequence of Lemma 7.16, since D and E are valid by construction.

Corollary 8.2

If $\exists t', A'$ *with* $A' \leq^{syn}_{S_\diamond} \langle A \rangle$, *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D_\mathcal{P} \cup E_\mathcal{P}} t'^{:\{A'\} \cup V}$	*then* $\mathcal{P} \vdash t : A$.
If $\exists t', A$ *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D_\mathcal{P} \cup E_\mathcal{P}} t'^{:\{A\} \cup V}$	*then* $\mathcal{P} \vdash EX\ t$.
If $\exists t_0, A$ *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D_\mathcal{P} \cup E_\mathcal{P}} t_0^{:\{A\} \cup U} \overset{*}{\longleftrightarrow}_{D \cup E} t'^{:\downarrow\emptyset}$	*then* $\mathcal{P} \vdash t = t'$.

Note that imposing that the set of decorations becomes non empty at the top is essential, as shown by the following example. For the presentation $\mathcal{P}$:

$$a : A,\ b : B,\ f : C \to C,\ a = b$$

using $\longleftrightarrow_{D_\mathcal{P} \cup E_\mathcal{P}}$, we get

$$f(a^{:\emptyset})^{:\emptyset} \rightarrowtail_{D_\mathcal{P}} f(a^{:\{A\}})^{:\emptyset} \longleftrightarrow_{E_\mathcal{P}} f(b^{:\{B\}})^{:\emptyset}$$

but $\mathcal{P} \vdash f(a) = f(b)$ is not a valid deduction, since there is no way to deduce that $f(a)$ exists in this G-algebra.

The second step of the proof consists of showing that every G-deduction can be mapped to some replacement of equal by equal on decorated terms, using decorated substitutions. We use an induction on G-deduction trees of the considered formulas (i.e. either $t = t'$, $t : A$ or $EX\ t$). Two deduction trees are compared using the lexicographic combination of the number of branching as first component, and the length of the derivation as second component. (If one prefers, the induction works also on the number of nodes in the deduction tree). Note that all terms in the constructed proofs are valid and the used substitutions are the same as in the proofs given by the induction hypothesis, i.e. they fulfill the conditions of decorated substitutions. In the following, we will simply write E instead of $E_\mathcal{P}$ and D instead of $D_\mathcal{P}$, in order to be more concise.

We have three different cases for the formulas to prove. Therefore, we distinguish three cases:

CASE A: $(\mathcal{P} \vdash t = t') \Rightarrow \exists t_0, A$ *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E} t_0^{:\{A\} \cup U} \overset{*}{\longleftrightarrow}_{D \cup E} t'^{:\downarrow\emptyset}$.
CASE B: $(\mathcal{P} \vdash t : A) \Rightarrow \exists t', A'$ *with* $A' \leq^{syn}_{S_\diamond} \langle A \rangle$ *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E} t'^{:\{A'\} \cup V}$.
CASE C: $(\mathcal{P} \vdash EX\ t) \Rightarrow \exists t'$ *s.t.* $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E} t'^{:\{A\} \cup V}$.

Remark that all terms $t^{:S}$ in the constructed proof satisfy $(t^{:S})^{:\downarrow\emptyset} \in \mathcal{T}(\Sigma, \mathcal{X})$ (written (H_1)) and $(t^{:S})^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_D t^{:S}$ (written (H_2)), a very useful property.
CASE A: We reason by case on the possible production of $t = t'$.

1. **Base case**
   If $t = t'$ is an equality in $\mathcal{P}$, then $t^{:\downarrow\emptyset} \rightarrowtail_D t^{:\{\Omega\}}$ by the rule $(t^{:s} \to t^{:s \cup \{\Omega\}}$ if $\{\Omega\} \not\subseteq s)$. Therefore, $t^{:\downarrow\emptyset} \rightarrowtail_D t^{:\downarrow\emptyset:\{\Omega\}} \leftarrowtail_D t^{:\downarrow\emptyset} \longleftrightarrow_E t'^{:\downarrow\emptyset}$. Note that all substitutions needed for the application of the rules always satisfy the conditions of decorated substitutions since variables are supposed to have their sort automatically in their decoration. Properties (H_1) and (H_2) are obviously satisfied.

   Otherwise $t = t'$ may be obtained from the application of one of the rule **Reflexivity**, **Symmetry**, **Transitivity**, **EqSubstitutivity** or **EqReplacement**. Let us consider each case.

2. **Reflexivity** $EX\ t \vdash t = t$.
   Since the deduction tree for $EX\ t$ is smaller, we can apply the induction hypothesis and thus

$$\mathcal{P} \vdash EX\ t \Rightarrow \exists t' \text{ s.t. } t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D \cup E} t'^{:\{A\} \cup V}$$

exists and satisfies $(H_i)_{i=1,2}$, which proves that

$$t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t'^{:\{A\}\cup V} \overset{*}{\longleftrightarrow}_{D\cup E} t^{:\downarrow\emptyset}.$$

The properties $(H_i)_{i=1,2}$ also hold for the constructed proof.

3. **Symmetry** $u = v \vdash v = u$.
Since the deduction tree for $u = v$ is smaller, we can apply the induction hypothesis and thus get the result which is symmetric.

4. **Transitivity** $u = v, v = w \vdash u = w$.
Since the deduction trees for $u = v$ and $v = w$ are smaller, we can apply twice the induction hypothesis and concatenate the equational proofs.

5. **EqSubstitutivity** $r = r' \vdash \sigma(r) = \sigma(r')$
with $t \equiv \sigma(r)$ and $t' \equiv \sigma(r')$ where $\equiv$ denotes syntactic identity.
In this case, by induction hypothesis

$$\mathcal{P} \vdash r = r' \Rightarrow \exists r_0, A \; s.t. \; r^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} r_0{}^{:\{A\}\cup U} \overset{*}{\longleftrightarrow}_{D\cup E} r'^{:\downarrow\emptyset}.$$

Note that to any $\mathcal{P}$-conform substitution in G-algebra that associates to $(x_i :: A_i)$ a term t_i for which there exists a proof $(t_i : A_i)$, we have $(t_i : A_i)$ as an implicit subproof for the preconditions of **EqSubstitutivity**. The fact that this subproof does not appear in the syntax of the calculus does not mean that it is not required for the application of the rule.We apply the induction hypothesis also on these membership proofs and have therefore:

$$\mathcal{P} \vdash t_i : A_i \Rightarrow \exists t_i', A_i' \leq^{syn}_{\mathcal{S}_\diamond} \langle A_i \rangle \; s.t. \; t_i^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t_i'^{:\{A_i'\}\cup U_i}$$

Remark that all substitutions in these proofs must be decorated substitutions, i.e. all terms in their ranges are valid. Consequently,

$$\begin{aligned}\exists r_0, A \; s.t. \; \sigma(r[t_i])^{:\downarrow\emptyset} &\overset{*}{\longleftrightarrow}_{D\cup E} \sigma(r)^{:\downarrow\emptyset}[t_i'^{:\{A_i'\}\cup U_i}] \overset{*}{\longleftrightarrow}_{D\cup E} \\ &\sigma(r')^{:\downarrow\emptyset}[t_i'^{:\{A_i'\}\cup U_i}] \overset{*}{\longleftrightarrow}_{D\cup E} \sigma(r'[t_i])^{:\downarrow\emptyset}.\end{aligned}$$

The properties $(H_i)_{i=1,2}$ follow from the induction hypothesis for $(r = r')$ and all $(t_i : A_i)$ together with stability of decoration rewriting by substitutivity, respectively context.

6. **EqReplacement** $t[u] = w, u = v \vdash t[v] = w$.
Both formulas $t[u] = w$ and $u = v$ are smaller than $t[v] = w$ and by applying the induction hypothesis we get:

$$\begin{aligned}&\exists t_0, A \; s.t. \; t[u]^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t_0{}^{:\{A\}\cup U} \overset{*}{\longleftrightarrow}_{D\cup E} w^{:\downarrow\emptyset} \\ \text{and} \quad &\exists t_1, B \; s.t. \; u^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t_1{}^{:\{B\}\cup U} \overset{*}{\longleftrightarrow}_{D\cup E} v^{:\downarrow\emptyset}.\end{aligned}$$

But since $\overset{*}{\longleftrightarrow}_{D\cup E}$ is stable under context, we can write:

$$\begin{aligned}t^{:\downarrow\emptyset}[v^{:\downarrow\emptyset}] \; &\overset{*}{\longleftrightarrow}_{D\cup E} \; t^{:\downarrow\emptyset}[t_1{}^{:\{B\}\cup U}] \\ &\overset{*}{\longleftrightarrow}_{D\cup E} \; t^{:\downarrow\emptyset}[u^{:\downarrow\emptyset}] =_d t[u]^{:\downarrow\emptyset} \\ &\overset{*}{\longleftrightarrow}_{D\cup E} \; t_0{}^{:\{A\}\cup U} \\ &\overset{*}{\longleftrightarrow}_{D\cup E} \; w^{:\downarrow\emptyset}.\end{aligned}$$

The properties $(H_i)_{i=1,2}$ are once more guaranteed by the induction hypothesis on $(t[u] = w)$ and $(u = v)$ together with stability of decoration rewriting by context.

CASE B: The proof goes on the same principle as for CASE A.

1. **Base case**
   If $t : A$ is a formula in $\mathcal{P}$, then $t^{:\downarrow\emptyset} \rightarrowtail_D t^{:\{A\}}$. Clearly, this guarantees properties $(H_i)_{i=1,2}$. Otherwise $t : A$ may be obtained from the application of one of the rules **Globality**, **VariableMembership**, **MeReplacement** or **MeSubstitutivity**.

2. **Globality** $EX\, t \vdash t : \Omega$.
   By induction we get $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t'^{:\{A\}\cup V}$. Then $A \leq^{syn}_{\mathcal{S}_\diamond} \langle\Omega\rangle$ and we are done. This trivially provides for properties $(H_i)_{i=1,2}$.

3. **VariableMembership** $x :: A \vdash x : A$.
   True because each variable is decorated by the sort given by its declaration. $(H_i)_{i=1,2}$ are trivial.

4. **MeReplacement** $t[u] : A, u = v \vdash t[v] : A$.
   In this case, we apply the induction hypothesis on the first premise and the result of CASE A on the second one. Thus we have:
   $\exists t', A' \leq^{syn}_{\mathcal{S}_\diamond} A$ *s.t.* $t[u]^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t'^{:\{A'\}\cup U}$
   and
   $\exists t_1, B$ *s.t.* $u^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t_1^{:\{B\}\cup U} \overset{*}{\longleftrightarrow}_{D\cup E} v^{:\downarrow\emptyset}$.
   But since $\overset{*}{\longleftrightarrow}_{D\cup E}$ is stable under context, we can write:
   $t^{:\downarrow\emptyset}[v^{:\downarrow\emptyset}] \overset{*}{\longleftrightarrow}_{D\cup E} t^{:\downarrow\emptyset}[t_1^{:\{B\}\cup U}] \overset{*}{\longleftrightarrow}_{D\cup E} t^{:\downarrow\emptyset}[u^{:\downarrow\emptyset}] =_d t[u]^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{D\cup E} t'^{:\{A'\}\cup U}$,
   which concludes this case.
   For $(H_i)_{i=1,2}$ see CASE A, subcase **EqReplacement**.

5. **MeSubstitutivity** $t : A \vdash \sigma(t) : A$.
   Applying the induction hypothesis to $t : A$ and all the $t_i : A_i$ necessary for the proof of $\mathcal{P} - conformity$ of σ, we get the conclusion, using the same construction as in the **EqSubstitutivity** - subcase of CASE A.

CASE C: This is the most complex case where we need to go upper back in the deduction tree. Let us begin with the easy cases:

1. **Base case**
   If $EX\, t$ is a formula in $\mathcal{P}$, then $t^{:\downarrow\emptyset} \rightarrowtail_D t^{:\{\Omega\}}$. Clearly, these proofs have the properties $(H_i)_{i=1,2}$. Otherwise $EX\, t$ may be obtained from the application of one **ExMembership**, **ExEquality**, **ExReplacement**, **ExSubstitutivity** or **ExSubterm**.

2. **ExMembership** $t : A \vdash EX\, t$.
   Applying the result of CASE B on $t : A$, we get the result.

3. **ExEquality** $t = t' \vdash EX\, t$.
   Applying the result of CASE A on $t = t'$, we extract the result.

4. **ExReplacement** $EX\, t[u], u = v \vdash EX\, t[v]$.
   Both formulas $EX\, t[u]$ and $u = v$ have a complexity smaller than $EX\, t[v]$ and by applying the

induction hypothesis we get:
$\exists t', A$ *s.t.* $t[u]^{:\downarrow\emptyset} \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} t'^{:\{A\}\cup U}$
and
$\exists t_1, B$ *s.t.* $u^{:\downarrow\emptyset} \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} t_1^{:\{B\}\cup U} \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} v^{:\downarrow\emptyset}$.
But since $\leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E}$ is stable under context, we can write:
$t^{:\downarrow\emptyset}[v^{:\downarrow\emptyset}] \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} t^{:\downarrow\emptyset}[t_1^{:\{B\}\cup U}] \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} t^{:\downarrow\emptyset}[u^{:\downarrow\emptyset}] =_d t[u]^{:\downarrow\emptyset} \leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E} t'^{:\{A\}\cup U}$
which concludes this case.
For $(H_i)_{i=1,2}$, see once more CASE A, subcase **EqReplacement**.

5. **ExSubstitutivity** $EX\ t \ \vdash \ EX\ \sigma(t)$.
Applying the induction hypothesis to $EX\ t$ and all the $t_i : A_i$ necessary for the proof of $\mathcal{P} - conformity$ of σ, we get the conclusion, using the same construction as in the **EqSubstitutivity** - subcase of CASE A.

6. **ExSubterm** $EX\ t[u] \ \vdash \ EX\ u$.

Here we need to consider more deeply the deduction tree.

(a) First, if $EX\ t[u]$ is a formula in $\mathcal{P}$, since u is a subterm of t, there exists a rule in D such that $u^{:\downarrow\emptyset} \rightarrowtail_D u^{:\{\Omega\}}$. Otherwise, $EX\ t[u]$ is itself obtained from an application of **ExSubstitutivity**, **ExSubterm**, **ExEquality**, **ExReplacement**, or **ExMembership**, respectively abbreviated by **ExSubst**, **ExS**, **ExE**, **ExR** and **ExM**.

(b) Assume that the previous step in the deduction tree is an application of **ExSubstitutivity**.
The deduction
$EX\ t[v] \vdash_{\mathbf{ExSubst}} EX\ \sigma(t)[\sigma(v)] \vdash_{\mathbf{ExS}} EX\ \sigma(v)$
where $u \equiv \sigma(v)$, can be transformed into
$EX\ t[v] \vdash_{\mathbf{ExS}} EX\ v \vdash_{\mathbf{ExSubst}} EX\ \sigma(v)$..
We can apply the induction hypothesis on $EX\ v$ in the second deduction tree since it is smaller by definition, and get the result using stability of $\leftrightarrow\!\!\!\!\diamond\!\!\!\!\rightarrow^*_{D\cup E}$ under substitution.

(c) If the previous step in the deduction tree is an application of **ExSubterm**, the deduction
$EX\ t'[t[u]] \vdash_{\mathbf{ExS}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$,
can be replaced by the shorter following one: $EX\ t'[t[u]] \vdash_{\mathbf{ExS}} EX\ u$, on which the induction hypothesis can be applied directly on $EX\ u$.

(d) Let us now consider the case where the deduction rule applied before **ExSubterm** is **ExEquality**. We are in the following situation:
$t[u] = t' \vdash_{\mathbf{ExE}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$.
If $t[u] = t'$ is an equality of the presentation, then u can be decorated by the universe sort Ω and $u^{:\downarrow\emptyset} \rightarrowtail_D u^{:\Omega}$.
Otherwise, we need to go one step further up in the deduction tree. This step may be an application of one of **Reflexivity**, **EqSubstitutivity** or **EqReplacement**. The two last names are abbreviated by **EqS** and **EqR**.

i. either
$EX\ t[u] \vdash_{\mathbf{Reflexivity}} t[u] = t' \vdash_{\mathbf{ExE}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $t' \equiv t[u]$. This can be transformed into

$EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
which is a shorter deduction and we get the result by induction.

ii. or
$t_0[u_0] = t'_0 \vdash_{\mathbf{EqS}} t[u] = t' \vdash_{\mathbf{ExE}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $t[u] \equiv \sigma(t_0[u_0])$ and $t' \equiv \sigma(t'_0)$. This can be transformed into
$t_0[u_0] = t'_0 \vdash_{\mathbf{ExE}} EX\ t_0[u_0] \vdash_{\mathbf{ExS}} EX\ u_0 \vdash_{\mathbf{ExSubst}} EX\ u$.
By induction hypothesis, there exists a proof for $EX\ u_0$ which is stable under σ and provides a proof for u.

iii. or the applied rule is **EqReplacement**. Depending of the localization of the redex we have the following two subcases:

A. either
$t[u'[v]] = t', v = w \vdash_{\mathbf{EqR}} t[u] = t' \vdash_{\mathbf{ExE}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $u \equiv u'[w]$. This can be transformed into
$t[u'[v]] = t' \vdash_{\mathbf{ExE}} EX\ t[u'[v]] \vdash_{\mathbf{ExS}} EX\ u'[v], v = w \vdash_{\mathbf{ExR}} EX\ u$
We apply the induction hypothesis on the first two steps of the new proof and continue then exactly in the same way as in the case of **ExReplacement** as last step of the $EX\ t$ proof.

B. or
$r[w] = t', w = u'[u] \vdash_{\mathbf{EqR}} r[u'[u]] = t' \vdash_{\mathbf{ExE}} EX\ r[u'[u]] \vdash_{\mathbf{ExS}} EX\ u$
where $t[u] \equiv r[u'[u]]$. Consider instead the deductions:
$w = u'[u] \vdash_{\mathbf{Symmetry}} u'[u] = w \vdash_{\mathbf{ExE}} EX\ u'[u] \vdash_{\mathbf{ExS}} EX\ u$.
Here the G-deduction tree is smaller because the whole branch concerning the deduction of $r[w] = t'$ has been dropped. The result holds by induction.

(e) or the rule applied before **ExSubterm** is **ExReplacement**. In this case, we have again two subcases depending on the position of the redex:

i. either
$EX\ t[u'[v]], v = w \vdash_{\mathbf{ExR}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $u \equiv u'[w]$. This is transformed into
$EX\ t[u'[v]] \vdash_{\mathbf{ExS}} EX\ u'[v], v = w \vdash_{\mathbf{ExR}} EX\ u$.
We can apply the induction hypothesis on $EX\ u'[v]$ and $u = v$ and by combining them, we get the conclusion.

ii. or
$EX\ r[w], w = r'[u] \vdash_{\mathbf{ExR}} EX\ r[r'[u]] \vdash_{\mathbf{ExS}} EX\ u$
where $t[u] \equiv r[r'[u]]$. Consider instead the deductions
$w = r'[u] \vdash_{\mathbf{Symmetry}} r'[u] = w \vdash_{\mathbf{ExE}} EX\ r'[u] \vdash_{\mathbf{ExS}} EX\ u$.
The G-deduction tree is smaller because the whole branch concerning the deduction of $EX\ r[w]$ has been dropped. The result holds by induction.

(f) Let us eventually consider the case where the deduction rule applied before **ExSubterm** is **ExMembership**. We are in the following situation:
$t[u] : A \vdash_{\mathbf{ExM}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$.
If $t[u] : A$ is an axiom of $\mathcal{P}$, there exists a rule in D such that $u^{:\downarrow\emptyset} \rightarrowtail_D u^{:\{\Omega\}}$. Otherwise, we need to go one step further up in the deduction tree. This step may be an application of **MeSubstitutivity** or **MeReplacement**, respectively abbreviated by **MeS** and **MeR**. It cannot be an application of **VariableMembership**, since $t[u]$ cannot be a variable.

i. either
$t_0[u_0] : A \vdash_{\mathbf{MeS}} t[u] : A \vdash_{\mathbf{ExM}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $t[u] \equiv \sigma(t_0[u_0])$. This can be transformed into

$t_0[u_0] : A \vdash_{\mathbf{ExM}} EX\ t_0[u_0] \vdash_{\mathbf{ExS}} EX\ u_0 \vdash_{\mathbf{ExSubst}} EX\ u$.
By induction hypothesis, there exists a proof for $EX\ u_0$ which is stable under σ and provides a proof for u.

ii. or the applied rule is **MeReplacement**, abbreviated **MeR**. Depending of the localization of the redex we have the following two subcases:

A. either
$t[u'[v]] : A, v = w \vdash_{\mathbf{MeR}} t[u'[w]] : A \vdash_{\mathbf{ExM}} EX\ t[u] \vdash_{\mathbf{ExS}} EX\ u$
where $u \equiv u'[w]$. This can be transformed into
$t[u'[v]] : A \vdash_{\mathbf{ExM}} EX\ t[u'[v]] \vdash_{\mathbf{ExS}} EX\ u'[v], v = w \vdash_{\mathbf{ExR}} EX\ u$

We apply the induction hypothesis on the first two steps of the new proof and continue then exactly as in the case of **ExReplacement** as last step in the $EX\ t$ proof.

B. or
$r[w] : A, w = u'[u] \vdash_{\mathbf{MeR}} r[u'[u]] : A \vdash_{\mathbf{ExM}} EX\ r[u'[u]] \vdash_{\mathbf{ExS}} EX\ u$
where $t[u] \equiv r[u'[u]]$. Consider instead the deductions:
$w = u'[u] \vdash_{\mathbf{Symmetry}} u'[u] = w \vdash_{\mathbf{ExE}} EX\ u'[u] \vdash_{\mathbf{ExS}} EX\ u$.
The G-deduction tree is smaller because the whole branch concerning the deduction of $r[w] : A$ has been dropped. The result holds by induction.

This concludes the proof of the theorem.

9 Confluence and Critical Pairs

We now consider the case where equalities can be ordered into rewrite rules. Our concern is to check under which conditions on the presentation, we can obtain a set of decoration and decorated rewrite rules that operationally are powerful enough to prove equational, existential and membership theorems. This is the case if any equational proof using $D \cup R$ has a rewrite proof, i.e. a proof without peaks.

The first goal of this section is to characterize confluence on a set $\mathcal{T}$ of valid terms. Defining notions on non-valid terms is obviously non-sense, since this would allow for incorrect interpretations using Theorem 8.1.

This can be exemplified by the following example:

Example 9.1 *Let $\mathcal{P}_{DER}$ be the following (sort inheriting) presentation associated to $(D, \emptyset, R)$ with*

$$\begin{aligned} D &= \{a^{:s} \to a^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s, \\ &\quad\ b^{:s} \to b^{:s \cup \{B\}} \text{ if } \{B\} \not\subseteq s, \\ &\quad\ f((x :: A)^{:\{A\}})^{:s} \to f((x :: A)^{:\{A\}})^{:s \cup \{C\}} \text{ if } \{C\} \not\subseteq s, \\ &\quad\ f((y :: B)^{:\{B\}})^{:s} \to f((y :: B)^{:\{B\}})^{:s \cup \{C\}} \text{ if } \{C\} \not\subseteq s\} \\ R &= \{f((x :: A)^{:\{A\}})^{:\{A\}} \to (x :: A)^{:\{A\}}, \\ &\quad\ f((y :: B)^{:\{B\}})^{:\{B\}} \to b^{:\{B\}}\} \end{aligned}$$

Clearly, $D \cup R$ is confluent on all valid terms, but the non-valid term $f(a^{:\{A,B\}})^{:\{C\}}$ rewrites to $a^{:\{A,B\}}$ respectively $b^{:\{B\}}$, which are both irreducible.

9.1 $\mathcal{T}$-Confluence

Well-known notions are redefined in a partial manner, i.e. with respect to a term set $\mathcal{T}$, as this was already done for unification.

Definition 9.2 *Let R be a set of decorated rewrite rules, D a set of decoration rules and $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$ a set of decorated terms.*

$D \cup R$ is called locally confluent on $\mathcal{T}$ *iff for all $t^{:T}, t_1^{:T_1}, t_2^{:T_2} \in \mathcal{T}$:*

$$t_1^{:T_1} \leftarrow_{D \cup R} t^{:T} \rightarrow_{D \cup R} t_2^{:T_2} \text{ implies } \exists t_0^{:T_0} \; : \; t_1^{:T_1} \stackrel{*}{\rightarrow}_{D \cup R} t_0^{:T_0} \stackrel{*}{\leftarrow}_{D \cup R} t_2^{:T_2}.$$

$D \cup R$ is called confluent on $\mathcal{T}$ *iff for all $t^{:T}, t_1^{:T_1}, t_2^{:T_2} \in \mathcal{T}$:*

$$t_1^{:T_1} \stackrel{*}{\leftarrow}_{D \cup R} t^{:T} \stackrel{*}{\rightarrow}_{D \cup R} t_2^{:T_2} \text{ implies } \exists t_0^{:T_0} \; : \; t_1^{:T_1} \stackrel{*}{\rightarrow}_{D \cup R} t_0^{:T_0} \stackrel{*}{\leftarrow}_{D \cup R} t_2^{:T_2}.$$

Newman's lemma still has its equivalent in the decorated case, when $\mathcal{T}$ is downward complete w.r.t. a reduction ordering, especially $\rightarrow_{D \cup R}$.

Lemma 9.3 *Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$ be a downward complete set of decorated terms w.r.t. $\rightarrow_{D \cup R}$. If $\rightarrow_{D \cup R}$ is terminating on $\mathcal{T}$ then the local confluence of $\rightarrow_{D \cup R}$ on $\mathcal{T}$ implies that $D \cup R$ is confluent on $\mathcal{T}$.*

Proof: The proof follows the standard one (e.g. [Hue80]). Assume $t^{:S} \in \mathcal{T}$ and

$$t_1^{:S_1} \stackrel{*}{\leftarrow}_{D \cup R} t^{:S} \stackrel{*}{\rightarrow}_{D \cup R} t_2^{:S_2}.$$

Let us show confluence by noetherian induction on $\leftarrow_{D \cup R}$. If $t_1^{:S_1} =_d t^{:S}$ or $t_2^{:S_2} =_d t^{:S}$, then confluence is trivial. Else we can cut off the first rewriting steps in each direction:

$$t_1^{:S_1} \stackrel{*}{\leftarrow}_{D \cup R} t_1'^{:S_1'} \leftarrow_{D \cup R} t^{:S} \rightarrow_{D \cup R} t_2'^{:S_2'} \stackrel{*}{\rightarrow}_{D \cup R} t_2^{:S_2}.$$

Since we have the local confluence on $\mathcal{T}$ and $t^{:S} \in \mathcal{T}$, we know that $t_1'^{:S_1'} \stackrel{*}{\rightarrow}_{D \cup R} t_0^{:S_0} \stackrel{*}{\leftarrow}_{D \cup R} t_2'^{:S_2'}$ for some $t_0^{:S_0}$ since $\mathcal{T}$ is downward complete. Together with the confluence on smaller terms $(t_1'^{:S_1'}, t_2'^{:S_2'})$ by induction hypothesis we get $t_1^{:S_1} \stackrel{*}{\rightarrow}_{D \cup R} t_1''^{:S_1''} \stackrel{*}{\leftarrow}_{D \cup R} t_0^{:S_0}$ and $t_0^{:S_0} \stackrel{*}{\rightarrow}_{D \cup R} t_2''^{:S_2''} \stackrel{*}{\leftarrow}_{D \cup R} t_2^{:S_2}$. Using once more the confluence on $t_0^{:S_0}$ gives us $t_1''^{:S_1''} \stackrel{*}{\rightarrow}_{D \cup R} t^{*:S^*} \stackrel{*}{\leftarrow}_{D \cup R} t_2''^{:S_2''}$. Consequently,

$$t_1^{:S_1} \stackrel{*}{\rightarrow}_{D \cup R} t^{*:S^*} \stackrel{*}{\leftarrow}_{D \cup R} t_2^{:S_2},$$

i.e. $D \cup R$ is also confluent on $\mathcal{T}$. □

A simple example for a $\rightarrow_{D \cup R}$-downward complete set is $\mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\diamond, \mathcal{F}, \mathcal{X}_\diamond)$, due to Lemma 7.14.

9.2 Definition and Properties of Critical Pairs

The rewrite rule version of the typing process has been designed for expressing the critical interactions between membership declarations and equalities in a convenient way. Since the first are now encoded into the decoration rewrite system D, this amounts to define adequate notions of critical pairs. Although the superposition mechanism used to compute all critical pairs is uniform, we distinguish between decorated critical pairs, that are equalities obtained by superposition inside a decorated rewrite rule of R, and decoration critical pairs, that are conditional equalities obtained by superposition inside a decoration rule of D. This last kind of conditional critical pair will give rise to a new decoration rule.

The main difference with the classical definitions is once more partiality.

Definition 9.4 Decorated critical pairs (obtained by superposition *into* decorated rules)
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $g^{:S_g} \to d^{:S_d}$ be a rewrite rule in R and $l^{:S_l} \to r^{:S_r}$ (resp. $l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s$) be a rewrite rule in R (resp. D) with disjoint sets of variables. The two rules overlap if there exists a position ω in the set of non-variable positions of $g^{:S_g}$, such that the decorated terms $g^{:S_g}{}_{|\omega}$ (with $Deco(g^{:S_g}{}_{|\omega}) = U$, $S_l \not\subseteq U$) and $l^{:S_l}$ (resp. $l^{:U}$) have the $\mathcal{T}$-complete, $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sound, non-empty set Ψ of strict decorated unifiers. Then for any $\psi \in \Psi$, the overlapped decorated term $\psi(g^{:S_g})$ produces the $\mathcal{T}$-decorated critical pair $(p^{:S_1} = q^{:S_2})$ where $q^{:S_2} =_d \psi(d^{:S_d})$ and $p^{:S_1} =_d \psi(g[r^{:S_r}]_\omega{}^{:S_g})$ (resp. $p^{:S_1} =_d \psi(g^{:S_g})[\psi(l^{:U})^{:U \cup S_l}]_\omega$).

In the particular case where $\omega = \Lambda$, we have $U = S_g$ and so the critical pairs are:

$$p^{:S_1} =_d \psi(r^{:S_r}) \text{ and } q^{:S_2} =_d \psi(d^{:S_d}) \text{ (resp. } p^{:S_1} =_d \psi(l)^{:S_g \cup S_l} \text{ and } q^{:S_2} =_d \psi(d^{:S_d})).$$

The set of such $\mathcal{T}$-decorated critical pairs is denoted by $CP(R,R)_{|\mathcal{T}}$ (resp. $CP(D,R)_{|\mathcal{T}}$).

Definition 9.5 Decoration critical pairs (obtained by superposition *into* decoration rules)
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $(g^{:s} \to g^{:s \cup S_g}$ if $S_g \not\subseteq s)$ be a rewrite rule in D and $l^{:S_l} \to r^{:S_r}$ be a rewrite rule in R (resp. $(l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s)$ a rewrite rule in D) with disjoint sets of variables. The two rules overlap if there exists a position ω in the set of non-variable positions of $g^{:s}$, such that the decorated terms $g^{:\emptyset}{}_{|\omega}$ (with $Deco(g^{:\emptyset}{}_{|\omega}) = U$, $S_l \not\subseteq U$) and $l^{:S_l}$ (resp. $l^{:U}$) have the $\mathcal{T}$-complete, $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sound, non-empty set Ψ of strict decorated unifiers. Then for any $\psi \in \Psi$, the overlapped decorated term $\psi(g)^{:s}$ produces the $\mathcal{T}$-decoration critical pair $(p^{:s} = q^{:s \cup S}$ if $S_g \not\subseteq s)$, where $q^{:s \cup S} =_d \psi(g)^{:s \cup S_g}$ and $p^{:s} =_d \psi(g[r^{:S_r}]_\omega)^{:s}$ (resp. $p^{:s} =_d \psi(g[l^{:U \cup S_l}]_\omega)^{:s}$).

The sets of such $\mathcal{T}$-decoration critical pairs are denoted by $CP(R,D)_{|\mathcal{T}}$ (resp. $CP(D,D)_{|\mathcal{T}}$). In the previous definition, we can restrict our attention to superposition at occurrences $\omega \neq \Lambda$ in the set of non-variable positions of g, for the case of superposition of a rule of R into a rule of D, since superpositions at the top occurrence are already handled in the definition of a decorated critical pair.

Example 9.6 *Let us consider an example of critical pairs between rules of D. Assuming $x :: A$, $y :: B$, the rules:*

$$\begin{aligned} f(b^{:\{B\}})^{:s} &\to f(b^{:\{B\}})^{:s \cup \{C\}} \text{ if } \{C\} \not\subseteq s \\ (x^{:\{A\}} + f(y^{:\{B\}})^{:\emptyset})^{:s} &\to (x^{:\{A\}} + f(y^{:\{B\}})^{:\emptyset})^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s \end{aligned}$$

overlap at position 2 of the second rule. This overlapping produces the critical pair:

$$(x^{:\{A\}} + f(b^{:\{B\}})^{:\{C\}})^{:s} = (x^{:\{A\}} + f(b^{:\{B\}})^{:\emptyset})^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s$$

Example 9.7 *Let us consider an example of superposition of a rule of R into a rule of D. Assuming $x :: A$, $y :: B$, the rule:*

$$f(y^{:\{B\}})^{:\{B\}} \to a^{:\{A,B\}}$$

in R overlaps the rule:

$$(x^{:\{A\}} + f(b^{:\{B\}})^{:\{B\}})^{:s} \to (x^{:\{A\}} + f(b^{:\{B\}})^{:\{B\}})^{:s \cup \{A\}} \text{ if } A \not\subseteq s$$

in D at position 2. This overlapping produces the critical pair:

$$(x^{:\{A\}} + a^{:\{A,B\}})^{:s} = (x^{:\{A\}} + f(b^{:\{B\}})^{:\{B\}})^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s.$$

Note that the right-hand side of this conditional critical pair is immediately reducible by the rule in R. The simplified pair can be oriented into a new rule of D:

$$(x^{:\{A\}} + a^{:\{A,B\}})^{:s} \to (x^{:\{A\}} + a^{:\{A,B\}})^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s.$$

To conclude this section, we prove the completeness of $\mathbf{UNIF_d}$ for the unification problems in the $CP_{\mathcal{T}}$'s, assuming the completeness over the set $strict\_subterm\_set(\mathcal{T})$ of all strict subterms of elements in $\mathcal{T}$. This is possible, since we do not superpose into variables.

Lemma 9.8 *Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ and let us assume that $terms(D \cup R) \subseteq Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. If $\mathbf{UNIF_d}$ is a $strict\_subterm\_set(\mathcal{T})$-complete algorithm, then it is also a $\mathcal{T}$-complete unification algorithm for all unification problems in $CP(R,R)_{|_\mathcal{T}}$, $CP(R,D)_{|_\mathcal{T}}$, $CP(D,D)_{|_\mathcal{T}}$, $CP(D,R)_{|_\mathcal{T}}$ and $CP(D,E)_{|_\mathcal{T}}$.*

Proof: Suppose that the unification problem has the form $x^{:S} \cong_d^? f(t_1,\ldots,t_n)^{:T}$. If $S \subsetneq T$, this is already a problem in solved form. Otherwise the two terms are not unifiable. $\widetilde{\mathbf{UNIF_d}}$ decides accordingly and returns $\{\sigma\}$ with $\sigma = \{x^{:S} \mapsto f(t_1,\ldots,t_n)^{:T}\}$ in the first case, which is trivially a complete solution, and $\emptyset$ in the second.

Since the definitions 9.5 and 9.4 do not allow variable overlaps, we must have in all other cases an initial unification problem $\mathcal{U}$ of the form $f(t_1,\ldots,t_n)^{:T} \cong_d^? g(t'_1,\ldots,t'_m)^{:T'}$. If $f \neq g$, $m \neq n$ or $T \not\approx T'$, then there is no solution, else the solutions of $\mathcal{U}$ in $\mathcal{T}$ are the same as those of $\bigwedge_{i\in[1..n]} t_i \cong_d^? t'_i$ in $strict\_subterm\_set(\mathcal{T})$, where $\mathbf{UNIF_d}$ is complete by assumption. □

9.3 Overlapping Computations

Let us now classify peaks of $D \cup R$. Let $t^{:V}, t'^{:S'}, t''^{:S''}$ be decorated terms such that:

$$t'^{:S'} \leftarrow^{\omega,\alpha,\phi'}_{D\cup R} t^{:V} \rightarrowtail^{\upsilon,\beta,\phi''}_{D\cup R} t''^{:S''},$$

with l and g being respectively the left-hand sides of ϕ' and ϕ''.

In the following it is useful to keep in mind that for any rule $l \to r$, $Deco(l) \subsetneq Deco(r)$ and therefore $Deco(\sigma(l)) \subsetneq Deco(\sigma(r))$ (since substitutivity is fulfilled by reduction orderings and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$) for any decorated substitution σ, as this is required by the definition of decorated rewrite rules and by construction for the decoration rules.

As usual for classifuing peaks, let us consider the classical relative positions of the redexes $t_{|\omega}$ and $t_{|\upsilon}$.

Disjoint case: υ and ω are incomparable positions.

Variable overlap case: we can assume without loss of generality that υ is the top position Λ in t. Thus $t \cong_d \beta(g)$ and there exists a variable x in g whose position in t is prefix of ω.

Critical overlap case: again we can assume without loss of generality that $\upsilon = \Lambda$. Then $t \cong_d \beta(g)$ and ω is a non-variable position in g.

Lemma 9.9 Variable overlap lemma
Let $t^{:V}, t'^{:S'}, t''^{:S''}$ be decorated terms such that $t^{:V}$ is valid and:

$$t'^{:S'} \leftarrow^{\omega,\alpha,\phi'}_{D\cup R} t^{:V} \rightarrowtail^{\Lambda,\beta,\phi''}_{D\cup R} t''^{:S''},$$

with l and g being respectively the left-hand sides of ϕ' and ϕ'', r being the right-hand side of ϕ' and ω being below a variable position in g. Then, the peak is convergent: $t'^{:S'} \downarrow_{D\cup R} t''^{:S''}$.

Proof: Let $x :: A$ be the variable of g, at position v' under which the rewrite step occurs. Then $\beta(x)$ contains the redex $\alpha(l)$ at some position v''. So $\omega = v'v''$.

Let σ be the decorated substitution defined by $\sigma(y) = \beta(y)$ if $y \neq x$ and $\sigma(x) = \beta(x)[\alpha(r)]_{v''}$. The last term is valid by Corollary 7.12. Furthermore, $Deco(r) \subsetneq Deco(l)$ and hence σ is also a decorated substitution. Since $\alpha(l) \cong_d t_{|\omega}$, the peak is convergent:

$$t'^{:S'} \overset{*}{\rightarrowtail}^{\alpha,\phi'}_{D \cup R} \sigma(g) \rightarrowtail^{\Lambda,\sigma,\phi''}_{D \cup R} \sigma(d) \overset{*}{\leftarrowtail}^{\alpha,\phi'}_{D \cup R} t''^{:S''}.$$

□

Lemma 9.10 *$\mathcal{T}$-critical pair lemma*
Let $t^{:V}, t'^{:S'}, t''^{:S''} \in \mathcal{T}$ be decorated terms such that:

$$t'^{:S'} \leftarrowtail^{\omega,\alpha,\phi'}_{D \cup R} t^{:V} \rightarrowtail^{\Lambda,\beta,\phi''}_{D \cup R} t''^{:S''}$$

with ω being a non-variable position in the left-hand side of ϕ''. Then, either there exists a $\mathcal{T}$-critical pair:

$$(p^{:S_1} = q^{:S_2}) \text{ if } \phi'' \in R, \text{ or } (p^{:s} = q^{:s \cup S} \text{ if } S \not\subseteq s) \text{ if } \phi'' \in D,$$

of the rule ϕ' on the rule ϕ'' at position ω, or the peak converges trivially:

$$t'^{:S'} \rightarrowtail^{\Lambda,\beta,\phi''}_{D \cup R} t_0^{:S_0} \leftarrowtail^{\omega,\alpha,\phi'}_{D \cup R} t''^{:S''}.$$

Moreover, if the peak is not trivially convergent, there is a decorated substitution ψ in a $\mathcal{T}$-complete set of decorated unifiers according to definitions 9.4, 9.5, such that $\beta\alpha \gtrsim^W_d \psi$ with $W = Var(g) \cup Var(l)$ and there exists a decorated substitution τ, such that $t'^{:S'} \cong_d \tau(p^{:S_1})$ and $t''^{:S''} \cong_d \tau(q^{:S_2})$ (resp. $t'^{:S'} \cong_d \tau(p^{:U})$ and $t''^{:S''} \cong_d \tau(p^{:S \cup U})$ for some $U \not\subseteq S$).

Proof: The proof is almost similar to the usual one [Hue80, JK86].
We prove the existence of ψ for all four cases. Let therefore ϕ' be of the form $l_1^{:S_{l_1}} \rightarrow r_1^{:S_{r_1}}$ if it is in R (resp. $(l_1^{:s} \rightarrow l_1^{:s \cup S_1}$ if $S_1 \not\subseteq s)$ if it is in D), and ϕ'' of the form $l_2^{:S_{l_2}} \rightarrow r_2^{:S_{r_2}}$ if it is in R (resp. $(l_2^{:s} \rightarrow l_2^{:s \cup S_2}$ if $S_2 \not\subseteq s)$ if it is in D).

- $\phi' \in R$, $\phi'' \in R$:
  Since $t^{:V} \cong_d \beta(l_2^{:S_{l_2}})$ and $t^{:V}{}_{|\omega} \cong_d \beta(l_2^{:S_{l_2}}{}_{|\omega}) \cong_d \alpha(l_1^{:S_{l_1}})$, by definition of decorated matching, and since the substitutions α and β can be supposed to have disjoint domains, $\alpha\beta$ is a decorated unifier of $l_2^{:S_{l_2}}{}_{|\omega} \cong^?_d l_1^{:S_{l_1}}$, i.e. in any $\mathcal{T}$-complete set of $\mathcal{D}$-unifiers we have a ψ with $\beta\alpha \gtrsim^W_d \psi$.
- $\phi' \in D$, $\phi'' \in R$:
  Let $Deco(t^{:V}{}_{|\omega}) = U$. Therefore, $t^{:V} \cong_d \beta(l_2^{:S_{l_2}})$ and $t^{:V}{}_{|\omega} \cong_d \beta(l_2^{:S_{l_2}}{}_{|\omega}) \cong_d \alpha(l_1^{:U})$. Consequently, $\alpha\beta$ is a decorated unifier of $l_2^{:S_{l_2}}{}_{|\omega} \cong^?_d l_1^{:U}$ and any $\mathcal{T}$-complete $\mathcal{D}$-unifer set must contain a ψ with $\beta\alpha \gtrsim^W_d \psi$. Furthermore U satisfies the conditions of ϕ' for s.
- $\phi' \in R$, $\phi'' \in D$:
  If $\omega = \Lambda$, then this case is already covered by the one with $\phi' \in D$ and $\phi'' \in R$. Else we have $t^{:V} \cong_d \beta(l_2^{:V})$ and $t^{:V}{}_{|\omega} \cong_d \beta(l_2^{:V}{}_{|\omega}) \cong_d \alpha(l_1^{:S_{l_1}})$. The existence of ψ can be obtained as before. Remark also, that V satisfies the conditions of ϕ'' for s.
- $\phi' \in D$, $\phi'' \in D$:
  If $\omega = \Lambda$ then the peak converges trivially. Else let $Deco(t^{:V}{}_{|\omega}) = U$. We have $t^{:V} \cong_d \beta(l_2^{:V})$ and $t^{:V}{}_{|\omega} \cong_d \beta(l_2^{:V}{}_{|\omega}) \cong_d \alpha(l_1^{:U})$. Therefore $\alpha\beta$ is a $\mathcal{T}$-decorated unifier of $l_1^{:U} \cong^?_d l_2^{:V}{}_{|\omega}$. Since $\omega \neq \Lambda$, $\alpha\beta$ must also be a decorated $\mathcal{T}$-unifier of $l_1^{:U} \cong^?_d l_2^{:\emptyset}{}_{|\omega}$. The existence of ψ follows now as before. Remark once more, that the conditions of ϕ' resp. ϕ'' are fulfilled by U resp. V.

The critical pairs can be constructed applying the definitions 9.4 and 9.5. Since $\psi \lesssim_d^{\alpha} \beta$, definitions 5.17, 6.8 and proposition 5.19 give us the existence of a substitution τ with $\tau(p^{:S_1}) \cong_d t'^{:S'}$ and $\tau(q^{:S_2}) \cong_d t''^{:S''}$ resp. $\tau(p^{:U}) \cong_d t'^{:S'}$ and $\tau(q^{:U \cup S}) \cong_d t''^{:S''}$. □

Definition 9.11 *A decorated critical pair* $(p^{:S_1} = q^{:S_2})$ *is* solved *if both decorated terms can be rewritten using* $\rightarrowtail_{D \cup R}$ *into the same decorated term. This is denoted* $(p^{:S_1} \downarrow_{D \cup R} q^{:S_2})$.

Definition 9.12 *A decoration critical pair* $(p^{:s} = q^{:s \cup S}$ if $S \not\subseteq s)$ *is* solved *if for any set of sorts* U *such that* $S \not\subseteq U$ $(p^{:U} \downarrow_{D \cup R} q^{:U \cup S})$.

Theorem 9.13 *Assume that* $\mathcal{P}$ *is a presentation given by a decorated rewrite system* R *and a set of decoration rules* D, *such that* $\rightarrowtail_{D \cup R}$ *is well-founded and* $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F})$ *is downward complete w.r.t.* $\rightarrowtail_{D \cup R}$.

If all $\mathcal{T}$*-critical pairs of* $D \cup R$ *are solved, then for any proof on decorated terms* $t'^{:S'} \overset{*}{\longleftrightarrow\!\!\!\diamond\!\!\!\rightarrowtail}_{D \cup R} t''^{:S''}$ *containing only terms in* $\mathcal{T}$, *there exists a rewrite proof:*

$$t'^{:S'} \overset{*}{\rightarrowtail}_{D \cup R} u^{:U} \overset{*}{\leftarrowtail}_{D \cup R} t''^{:S''}.$$

Proof: Consider the proof P:

$$t' = t_0 \longleftrightarrow\!\!\!\diamond\!\!\!\rightarrowtail_{D \cup R} t_1 \ldots t_{k-1} \longleftrightarrow\!\!\!\diamond\!\!\!\rightarrowtail_{D \cup R} t_k \longleftrightarrow\!\!\!\diamond\!\!\!\rightarrowtail_{D \cup R} t_{k+1} \ldots \longleftrightarrow\!\!\!\diamond\!\!\!\rightarrowtail_{D \cup R} t_n = t''.$$

Then either P is a rewrite proof or it contains a peak:

$$t_{k-1} \overset{\omega,\alpha,\phi'}{\longleftarrowtail}_{D \cup R} t_k \overset{\upsilon,\beta,\phi''}{\rightarrowtail}_{D \cup R} t_{k+1}.$$

Assume there is a peak. We prove by noetherian induction, using the rewrite relation as reduction ordering, that there is a rewrite proof without peak. We are in one of the following cases, according to the relative positions of the redexes $t_{k|_\omega}$ and $t_{k|_\upsilon}$.

Disjoint case: Then the two reductions commute:

$$t_{k-1} \overset{*}{\rightarrowtail}^{\upsilon,\beta,\phi''}_{D \cup R} u \overset{*}{\leftarrowtail}^{\omega,\alpha,\phi'}_{D \cup R} t_{k+1},$$

since the sort information that are used are independent. If t_{k-1} or t_{k+1} is at the top of a new peak, we can use the induction hypothesis, since both terms are smaller than t_k, implying that they are by definition also in $\mathcal{T}$.

Variable overlap case: Using Lemma 9.9, the peak is convergent: $t_{k-1} \downarrow_{D \cup R} t_{k+1}$. If t_{k-1} or t_{k+1} are then at the top of a new peak, we can use once more the induction hypothesis, as in the first case. Remark that, as before, t_{k-1}, t_{k+1} are in $\mathcal{T}$.

Critical overlap case: If it is a *decorated* critical pair, since all decorated $\mathcal{T}$-critical pairs are solved and by stability of the rewrite relation by substitution and context proved in Proposition 7.17, we have $t_{k-1} \overset{*}{\rightarrowtail}_{D \cup R} u \overset{*}{\leftarrowtail}_{D \cup R} t_{k+1}$. Since $t_{k-1}, t_{k+1}, t'_{k-1}, t'_{k+1}$ are all smaller than t_k, we have by induction hypothesis a rewrite proof, thanks to downward completeness of $\mathcal{T}$.

Assume now that this peak corresponds to a *decoration* critical pair $(p^{:s} = q^{:s \cup S_0}$ if $S_0 \not\subseteq s)$. Since all decoration $\mathcal{T}$-critical pairs are solved, this implies again that $t_{k-1} \overset{*}{\rightarrowtail}_{D \cup R} u \overset{*}{\leftarrowtail}_{D \cup R} t_{k+1}$, giving us as before the existence of a rewrite proof.

□

Let us now consider how to solve critical pairs in order to eliminate some easy cases.

Proposition 9.14 *Any decoration critical pair between decoration rules in D at an occurrence $\omega \neq \Lambda$ is solved by adding an enriched version of the upper decoration rule. Decoration critical pairs at Λ are always solved.*

Proof: Consider the decoration critical pair

$$(\psi(g[l^{:U\cup S_l}]_\omega)^{:s} = \psi(g)^{:s\cup S_g} \text{ if } S_g \not\subseteq s),$$

produced by overlapping the rule

$$\phi_1 \ : \ (l^{:s'} \to l^{:s'\cup S_l} \text{ if } S_l \not\subseteq s')$$

of D into the rule

$$\phi_2 \ : \ (g^{:s} \to g^{:s\cup S_g} \text{ if } S_g \not\subseteq s)$$

in D. Then $q^{:s\cup S}$ contains at position ω the subterm $\psi(g^{:\emptyset}{}_{|\omega}) =_d \psi(g^{:\emptyset}{}_{|\omega})^{:U}$ and $\psi(g^{:\emptyset}{}_{|\omega})^{:U} \cong_d \psi(l^{:U})$, since ψ is a $\mathcal{D}$-unifier of these two decorated terms. Thus the rule ϕ_2 of D applies on $q^{:s\cup S}$ and yields $\psi(g[l^{:U\cup S_l}]_\omega)^{:s\cup S_g}$. Assume now that the decoration rule

$$\phi \ : \ \psi(g[l^{:U\cup S_l}]_\omega)^{:s} \to \psi(g[l^{:U\cup S_l}]_\omega)^{:s\cup S_g} \text{ if } S_g \not\subseteq s$$

is added in D. Then for any set of sorts U' such that $S_g \not\subseteq U'$,

$$\psi(g[l^{:U\cup S_l}]_\omega)^{:U'} \rightarrowtail^{\phi}_{D\cup R} \psi(g[l^{:U\cup S_l}]_\omega)^{:U'\cup S_g}.$$

The critical pair is thus solved.

Superposition at Λ of two rules of D gives $p^{:s} = \psi(g)^{:s}$ and $q^{:s\cup S} = \psi(g)^{:s\cup S_g}$, which is solved in an obvious way using once more ϕ_2. □

Proposition 9.15 *Any decorated critical pair in $CP(D,R)$ obtained by overlapping a decoration rule into a decorated rewrite rule is solved by adding an enriched version of the decorated rewrite rule and a new decoration rule.*

Proof: The critical pair obtained by overlapping $(l^{:s} \to l^{:s\cup S_l}$ if $S_l \not\subseteq s)$ into $(g^{:S_g} \to d^{:S_d})$ using a strict decorated unifier ψ is solved in the case of $\omega \neq \Lambda$ by adding

$$\psi(g^{:S_g})[\psi(l^{:U})^{:U\cup S_l}]_\omega) \to \psi(d^{:S_d})$$

where $U = Deco(g^{:S_g}{}_{|\omega})$ and in the case of $\omega = \Lambda$ by adding

$$\psi(g^{:S_g})[\psi(l^{:S_g})^{:S_g\cup S_l}]_\Lambda) \to \psi(d^{:S_d})^{:S_d\cup S_g\cup S_l}$$

and

$$\psi(d)^{:s} \to \psi(d)^{:s\cup S_g\cup S_l\setminus S_d} \text{ if } S_g \cup S_l \setminus S_d \not\subseteq s.$$

Remark that $g \notin \mathcal{X}_\diamond$ if $\omega = \Lambda$, since we do not calculate critical pairs on variable positions. □

Note that $S_g \subsetneq S_d$ by the definition of decorated rewrite rules. Therefore, the new rule also represents a well-formed decorated rewrite rule in the case of $\omega \neq \Lambda$. In practice we take the critical pair as equality and normalize it first, of course, before we orient it into a rule. This may let disappear the decoration subsumption property and so it becomes necessary to add a new decoration rule when the subsumption property isn't fulfilled (see section 10.2 for details), using the same construction as above. The problem of solving decoration critical pairs may seem complex at a first glance, but becomes quite obvious by looking more closely at their form (see Examples 9.6 and 9.7).

Proposition 9.16 *Any decoration critical pair from R into D can be solved by adding a decoration rule.*

Proof: Consider the decoration critical pair

$$(\psi(g[r^{:S_r}]_\omega)^{:s} = \psi(g)^{:s \cup S_g} \text{ if } S_g \not\subseteq s),$$

produced by overlapping the rule

$$\phi_1 \ : \ l^{:S_l} \to r^{:S_r}$$

of R in the rule

$$\phi_2 \ : \ (g^{:s} \to g^{:s \cup S_g} \text{ if } S_g \not\subseteq s)$$

in D. Then $\psi(g)^{:s \cup S_g}$ contains at position $\omega \neq \Lambda$ the subterm $\psi(g^{:\emptyset}|_\omega) =_d \psi(g^{:\emptyset}|_\omega)^{:U}$ and $\psi(g^{:\emptyset}|_\omega)^{:U} \cong_d \psi(l^{:S_l})$, since ψ is a $\mathcal{D}$-unifier of these two decorated terms. Thus the rule ϕ_1 of R applies on $\psi(g)^{:s \cup S_g}$ and yields $\psi(g[r^{:S_r}]_\omega)^{:s \cup S_g}$. Now for any set of sorts U' such that $S_g \not\subseteq U'$,

$$\psi(g[r^{:S_r}]_\omega)^{:U'} \rightarrowtail^{\phi}_{D \cup R} \psi(g[r^{:S_r}]_\omega)^{:U' \cup S_g}$$

using the rule:

$$\phi \ : \ (\psi(g[r^{:S_r}]_\omega)^{:s} \to \psi(g[r^{:S_r}]_\omega)^{:s \cup S_g} \text{ if } S_g \not\subseteq s).$$

The critical pair is thus solved by adding to D this rule obtained by a straightforward orientation of the reduced critical pairs. □

Notation: $\overline{CP(R,D)}$ and $\overline{CP(D,D)}$ stand for the sets of decoration rules used to solve these critical pairs. $\overline{CP(\phi,D)}$ stands for the set of decoration rules generated by some rule $\phi \in D \cup R$.

10 Decorated Completion in Sort Inheriting presentations

10.1 General Purpose of a Completion Process

Our purpose is now to design a completion process that provides, whenever it does not fail, from an initial sort inheriting presentation $\mathcal{P}_0$, a saturated presentation $\mathcal{P}_\infty$ such that $Th(\mathcal{P}) = Th(\mathcal{P}_\infty)$ and any formula $\phi \in Th(\mathcal{P}_\infty)$ has a rewrite proof. Our notations are consistent with [Bac91].

To $\mathcal{P}_0$ is associated the initial triple (D_0, E_0, R_0) where E_0 is usually the set of decorated equalities $E_{\mathcal{P}_0}$, D the set of decoration rules $D_{\mathcal{P}_0}$ associated to $\mathcal{P}_0$ as defined in section 7.5 and R_0 is empty.
Notation: We write $t^{:S} \rightarrowtail_{\mathcal{P}} t'^{:S'}$ instead of $t^{:S} \ (\rightarrowtail_{R \cup D}) \ t'^{:S'}$.

The completion process is defined as a transformation rule system $\mathcal{OSC}$ that transforms decorated presentations: $\mathcal{P}_0 \vdash_{\mathcal{OSC}} \ldots \vdash_{\mathcal{OSC}} \mathcal{P}_k \ldots$.

The resulting decorated presentation, given by $(D_\infty, E_\infty, R_\infty)$ where $E_\infty = \emptyset$, satisfies the following properties, if the completion does not fail:

- The Church-Rosser property of $\mathcal{P}_\infty$: any equational theorem $(t = t')$ has a rewrite proof using $\mathcal{P}_\infty$, i.e. there exists a term t'' decorated by at least one sort A, such that:

$$t^{:\downarrow\emptyset} \rightarrowtail^*_{\mathcal{P}_\infty} t''^{:\{A\} \cup S''} \leftarrowtail^*_{\mathcal{P}_\infty} t'^{:\downarrow\emptyset},$$

  where S'' is a decoration.

- The type completeness property of $\mathcal{P}_\infty$: any membership theorem $(t : A)$, where $A \in \mathcal{S}$, has a rewrite proof using $\mathcal{P}_\infty$, i.e. there exists t' and $A' \in \mathcal{S}_\emptyset$ with $A' \leq^{syn}_{\mathcal{S}_\emptyset} (A)$, such that:

$$t^{:\downarrow\emptyset} \rightarrowtail^*_{\mathcal{P}_\infty} t'^{:\{A'\} \cup S'},$$

  where S' is a decoration.

- The existential completeness property of $\mathcal{P}_\infty$: any existential theorem $(EX\ t)$ has a rewrite proof using $\mathcal{P}_\infty$, i.e. there exist t' and $A \in \mathcal{S}_\emptyset$ such that:

$$t^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{\mathcal{P}_\infty} t'^{:\{A\}\cup S'},$$

where S' is an arbitrary decoration.

Since $E_\infty = \emptyset$, $\overset{*}{\rightarrowtail}_{\mathcal{P}_\infty}$ is actually rewriting with decoration rules in D_∞ and decorated rules in R_∞.

The dual aspect of completion is the proof reduction process. Let us make this notion more precise. Using the completeness Theorem 8.1, proofs of interest in a decorated presentation $\mathcal{P}$ given by (D, E, R) are of the form $t^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D\cup E} t_0^{:\{A\}\cup U} \overset{*}{\leftrightarrow}_{D\cup E} t'^{:\downarrow\emptyset}$. The set of completion rules is mirrored by a a set of proof reduction rules which normalizes such a proof into a rewrite proof $t^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{\mathcal{P}_\infty} t''^{:\{A\}\cup S''} \overset{*}{\leftarrowtail}_{\mathcal{P}_\infty} t'^{:\downarrow\emptyset}$.

10.2 Transformation Rules for Order-Sorted Completion

Equalities are ordered according to a given decorated reduction ordering $>_d$ on decorated terms (see Definition 7.21). Rewrite rules from R and from D are compared by the following ordering, derived from [DJ90]:

Definition 10.1 *The ordering on rewrite rules $\gg$ is defined by:*

$$l^{:S_l} \to r^{:S_r} \gg g^{:S_g} \to d^{:S_d}$$

if after first replacing sort variables by the empty set of sorts then:

- $l^{:S_l} \trianglelefteq g^{:S_g}$ *(a subterm of $l^{:S_l}$ is an instance of $g^{:S_g}$ modulo sort inheritance and not conversely),*
- *or else $l^{:S_l}$ and $g^{:S_g}$ are subsumption equivalent ($l^{:S_l} \lesssim_d^? g^{:S_g}$ and $g^{:S_g} \lesssim_d^? l^{:S_l}$) and $r^{:S_r} >_d d^{:S_d}$ in the given reduction ordering.*

The completion procedure is expressed with the set $\mathcal{OSC}$ of transformation rules given in Figure 8.

First of all, note that $\mathcal{OSC}$ only generates valid decorated terms, rules and equalities in each (D_k, E_k, R_k).

Lemma 10.2 *Let $(D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation starting with $D_0 = D_\mathcal{P}$, $E_0 = E_\mathcal{P}$ and $R_0 = \emptyset$.*

Then for all $k \geq 0$, all terms, all decorated rules and equalities in (D_k, E_k, R_k) are valid in $\mathcal{P}$.

Proof: The proof is an induction on k. For $k = 0$, obviously all terms $t^{:S}$ in $D_\mathcal{P} \cup E_\mathcal{P}$ satisfy $(t^{:S})^{:\downarrow\emptyset} =_d t^{:S}$ and are therefore trivially valid. The equations are in one-to-one connection with those in $\mathcal{P}$ and therefore also valid.

For $k > 0$, we get the validity of the terms in (D_k, E_k, R_k) by the fact that they either can be reached from terms in $(D_{k-1}, E_{k-1}, R_{k-1})$ by $\leftrightarrowtail_{D_{k-1},E_{k-1},R_{k-1}}$ or result from the application of a unifier for two valid terms and $\rightarrowtail_{D_{k-1},E_{k-1},R_{k-1}}$, or have a new sort in its top decoration, that was at the top of a term reachable by $\rightarrowtail_{D_{k-1},E_{k-1},R_{k-1}}$ in the last presentation. Consequently, these terms are also valid, because of Lemma 7.11, used together with the induction hypothesis, and the fact that decorated unification is subterm conservative.

The validity of the decorated rewrite rules and equalities in (D_k, E_k, R_k) is now a consequence of **EqReplacement**, **Symmetry** and **EqSubstitutivity**. Validity of decoration rewrite rules follows immediately from the validity of all terms. □

1. **Orient_SD**
$$\frac{D, E \cup \{p^{:S} = q^{:S'}\}, R}{D, E, R \cup \{p^{:S} \to q^{:S'}\}} \quad \text{if} \quad p^{:S} >_d q^{:S'} \text{ and } S \subsetneq S'$$

2. **Orient_NSD**
$$\frac{D, E \cup \{p^{:S} = q^{:S'}\}, R}{D \cup \{(q^{:s} \to q^{:s \cup S \setminus S'} \text{ if } S \setminus S' \not\subseteq s)\}, E, R \cup \{p^{:S} \to q^{:S \cup S'}\}}$$
if $p^{:S} >_d q^{:S \cup S'}$ and $S \not\subseteq S'$
and if $q \in \mathcal{X}_\diamond$ with $q :: A$ then $\{A\} \approx S \cup S'$

3. **Deduce**
$$\frac{D, E, R}{D, E \cup \{(p^{:S} = q^{:S'})\}, R} \quad \text{if } (p^{:S} = q^{:S'}) \in CP(R, R) \cup CP(D, R)$$

4. **Deduce_deco**
$$\frac{D, E, R}{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } S \not\subseteq s)\}, E, R} \quad \text{if } (p^{:s} = p^{:s \cup S} \text{ if } S \not\subseteq s) \in \overline{CP(R, D)} \cup \overline{CP(D, D)}$$

5. **Simplify**
$$\frac{D, E \cup \{(p^{:S} = q^{:S'})\}, R}{D, E \cup \{(p''^{:S''} = q^{:S'})\}, R} \quad \text{if } p^{:S} \rightarrowtail^{\sigma,\phi}_{D \cup R} p''^{:S''}$$

6. **Delete**
$$\frac{D, E \cup \{(p^{:S} = q^{:S'})\}, R}{D, E, R} \quad \text{if } p^{:S} \cong_d q^{:S'}$$

7. **Compose**
$$\frac{D, E, R \cup \{(l^{:S_l} \to r^{:S_r})\}}{D, E, R \cup \{(l^{:S_l} \to r'^{:S_{r'}})\}} \quad \text{if } r^{:S_r} \rightarrowtail^{\sigma,\phi}_{D \cup R} r'^{:S_{r'}}$$

8. **Compose_D_deco**
$$\frac{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } S \not\subseteq s)\}, E, R}{D \cup \{(p'^{:s} \to p'^{:s \cup S'} \text{ if } S' \not\subseteq s)\}, E, R} \quad \text{if } p^{:S} \rightarrowtail^{\omega,\sigma,\phi}_{D} p'^{:S'}$$

9. **Compose_R_deco**
$$\frac{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } S \not\subseteq s)\}, E, R}{D \cup \{(p'^{:s} \to p'^{:s \cup S} \text{ if } S \not\subseteq s)\}, E, R} \quad \text{if } p^{:S} \rightarrowtail^{\omega,\sigma,\phi}_{R} p'^{:S} \text{ and } \omega \neq \Lambda$$

10. **Subsume_deco**
$$\frac{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } c(s))\}, E, R}{D, E, R} \quad \text{if} \quad p^{:\emptyset} \rightarrowtail^{\Lambda,\sigma,\phi}_{D} p^{:S'} \text{ with } S \subsetneq S' \text{ and } (p^{:s} \to p^{:s \cup S} \text{ if } c(s)) \neq \phi$$

11. **Collapse**
$$\frac{D, E, R \cup \{(l^{:S_l} \to r^{:S_r})\}}{D, E \cup \{(l'^{:S_{l'}} = r^{:S_r})\}, R} \quad \text{if } l^{:S_l} \rightarrowtail^{\sigma,\phi}_{D \cup R} l'^{:S_{l'}} \;\&\; l^{:S_l} \to r^{:S_r} \gg \phi$$

Figure 8: *OSC* The completion rules for decorated terms.

Correctness of completion amounts to prove that any formula provable in a decorated presentation $\mathcal{P}$ obtained during the completion is equivalently provable in $\mathcal{P}_0$, the initial presentation.

Proposition 10.3 *The transformation rules in OSC are sound w.r.t. deduction in G-algebra. In other words, if the completion starts with* $\mathcal{P}_0 = (D_\mathcal{P}, E_\mathcal{P}, \emptyset)$ *and* $\mathcal{P}_0 \vdash_{OSC} \ldots \vdash_{OSC} \mathcal{P}_k$, *then* $Th(\mathcal{P}_0) = Th(\mathcal{P}_k)$.

Proof: We show the following property $\mathbb{H}$ for all $\mathcal{P}_k$, $k \geq 0$, by induction on k: for any terms t, t', t'',

1. $(\ t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{\mathcal{P}_k} t''^{:S''\cup\{A\}} \overset{*}{\longleftrightarrow}_{\mathcal{P}_k} t'^{:\downarrow\emptyset})$ iff $t = t' \in Th(\mathcal{P}_0)$,
2. $(\exists A' \leq^{syn}_{S_\diamond} \langle A\rangle : t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{\mathcal{P}_k} t'^{:S'\cup\{A'\}})$ iff $(t : A) \in Th(\mathcal{P}_0)$,
3. $(t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_{\mathcal{P}_k} t'^{:S'\cup\{A\}})$ iff $(EX\ t) \in Th(\mathcal{P}_0)$.

The base case is a consequence of Theorem 8.1, since $\mathcal{P}_k = \mathcal{P}_0$. The induction step from $\mathcal{P}_k$ to $\mathcal{P}_{k+1}$ assumes the equivalence for $\mathcal{P}_k$ and shows then the same equivalence for $\mathcal{P}_{k+1}$.

We denote $\mathcal{P}_k$ simply $\mathcal{P} = (D, E, R)$ and $\mathcal{P}_{k+1}$ simply $\mathcal{P}' = (D', E', R')$.

The first direction of the equivalence ($\Rightarrow$) results from Lemma 10.2 that guarantees the validity of all decorated rewrite rules, equalities and decoration rewrite rules. Hence, we can apply Lemma 7.16 in order to get $Th(\mathcal{P}') \subseteq Th(\mathcal{P}_0)$.

The second direction of the equivalence ($\Leftarrow$) is proved by transforming any proof of $\mathbb{H}$ in $\mathcal{P}$ into a proof of the same formula in $\mathcal{P}'$. Thus $Th(\mathcal{P}) \subseteq Th(\mathcal{P}')$ and by induction hypothesis $Th(\mathcal{P}) = Th(\mathcal{P}_0)$, which implies $Th(\mathcal{P}_0) \subseteq Th(\mathcal{P}')$.

This is shown using a proof reduction relation $\Longrightarrow$, that is also used in Section 10.3. We give to the proof reduction rules the same name as the completion rules except that they are underlined. In the sequel of the proof, we write $0,1$ over the $\rightarrowtail$ in order to represent one or none application of rewriting. For any of the following proof reduction rules, we assume of course the conditions of the corresponding completion rules to be fulfilled.

1. Orient_(N)SD:
   If $q \in \mathcal{X}_\diamond$, we can be sure that for all $B \in S \cup S'$, if $(q :: A) \in \mathcal{P}$, then $A \leq^{syn}_S B$, since $\{A\} \approx S \cup S'$, i.e. $S \setminus S' \not\subseteq s$ is unsatisfiable and the newly added decoration rule can be ignored.
   Suppose that the equality $p^{:S} = q^{:S'}$ is applicable at some term t at occurrence ω using σ, yielding t'. Then $t_{|\omega} \cong_d \sigma(p^{:S})$ and $t'_{|\omega} \cong_d \sigma(q^{:S'})$ by Definitions 6.1 and 7.2, thus the proof reduction is defined by:
   $$(t[\sigma(p^{:S})] \cong_d t \longleftrightarrow^{\sigma,p^{:S}=q^{:S'}}_{E} t' \cong_d t[\sigma(q^{:S'})]) \Longrightarrow (t \rightarrowtail^{\sigma,p^{:S}\to q^{:S\cup S'}}_{R'} t'' \overset{0,1}{\longleftarrow}^{\sigma,\phi}_{D'} t'),$$
   where – ϕ is $q^{:s} \to q^{:s\cup S\setminus S'}$ if $S \setminus S' \not\subseteq s$,
   – $t'' \cong_d t[\sigma(q^{:S'})^{:U}]$ (by Proposition 7.18) and
   – $U = Deco(\sigma(q^{:S'})) \cup S \setminus S'$.
   Remark that the equality symbol is commutative and therefore the inverse application of a rule can be transformed analogously.
2. Deduce:
   $$(t' \longleftarrow^{\sigma,\phi}_{R\cup D} t \rightarrowtail^{\sigma',\phi'}_{R} t'') \Longrightarrow (t' \longleftrightarrow^{\nu,\tau,p^{:S}=q^{:S'}}_{E'} t'')$$
   where τ is defined by $\sigma \circ \sigma'(t) \cong_d \tau \circ \psi(t)$ for some unifier ψ of the left hand sides of ϕ and ϕ' according to the definitions of $CP(R, R)$ and $CP(D, R)$.
   Remark that $\sigma \circ \sigma'$ and τ are decorated substitutions by Corollary 5.16 respectively Definitions 5.17 and 6.8. In the case of $CP(D, E)$, no transformation is necessary.

3. Deduce_deco:

$$(t' \leftarrow^{\sigma',\phi'}_{D\cup R} t \rightarrowtail^{\sigma,\phi}_{D} t'') \Longrightarrow (t' \rightarrowtail^{\tau,(p^{:s}\rightarrow p^{:s\cup S} \text{ if } S\not\subseteq s)}_{D'} t'_0 \overset{0,1}{\leftarrow}^{\sigma',\phi'}_{D'\cup R'} t''),$$

where τ is defined similarly as before by $CP(R,D)$ resp. $CP(D,D)$.

4. Simplify:
First of all $\phi \in D' \cup R'$ since $D \cup R = D' \cup R'$. Since $p^{:S} = q^{:S'}$ is applicable to t, we know that σ' (see below) is a decorated substitution for t and the application of ϕ to $p^{:S}$ guarantees us the same for σ. Furthermore the applicability of $p^{:S} = q^{:S'}$ to t implies the condition over the decorations of the variables in $\mathcal{R}an(\sigma)$ needed for Proposition 5.15, which now says that $\sigma' \circ \sigma$ is a decorated substitution. Let $\phi : g^{:S_g} \rightarrow d^{:S_d}$ be in R. Then:

$$(t[\sigma'(p^{:S}[\sigma(g^{:S_g})])] \cong_d t[\sigma'(p^{:S})] \cong_d t \longleftrightarrow^{\sigma',p^{:S}=q^{:S'}}_{E} t' \cong_d t[\sigma'(q^{:S'})])$$
$$\Longrightarrow$$
$$(t \rightarrowtail^{\sigma'\circ\sigma,\phi}_{R'} t'' \longleftrightarrow^{\sigma',p''^{:S''}=q^{:S'}}_{E'} t')$$

where – $p''^{:S''} \cong_d p^{:S}[\sigma(d^{:S_d})]$ and
– $t'' \cong_d t[\sigma'(p^{:S}[\sigma(d^{:S_d})])]$ by Proposition 7.18.

Now let $\phi : (l^{:s} \rightarrow l^{:s\cup S_l} \text{ if } S_l \not\subseteq s) \in D$. We define therefore:

$$(t[\sigma'(p^{:S}[\sigma(l^{:U})]_\omega)]_\nu \cong_d t[\sigma'(p^{:S})]_\nu \cong_d t \longleftrightarrow^{\sigma',p^{:S}=q^{:S'}}_{E} t' \cong_d t[\sigma'(q^{:S'})]_\nu)$$
$$\Longrightarrow$$
$$(t \rightarrowtail^{\sigma'\circ\sigma,\phi}_{D'} t'' \longleftrightarrow^{\sigma'\circ\sigma,p''^{:S''}=q^{:S'}}_{E'} t')$$

where – $t'' \cong_d t[\sigma'(p^{:S})]_\nu[\sigma' \circ \sigma(l^{:U})^{:U'\cup S_l}]_{\nu.\omega}$ (by Proposition 7.18),
– $U' = Deco(\sigma'(\sigma(l^{:U})))$ and
– $p''^{:S''} =_d p^{:S}[\sigma(l^{:U})^{:U'\cup S_l}]$.

5. Delete:

$$(t \longleftrightarrow^{p^{:S}=q^{:S'}}_{E} t) \Longrightarrow \Phi$$

where Φ denotes the empty proof.

6. Compose:
As in the case of **Simplify** we know that σ, σ' and $\sigma' \circ \sigma$ are decorated substitutions. Furthermore, ϕ is in $D' \cup R'$, since no rule can be composed with itself (since this would be in contradiction to the termination of the rules in $D \cup R$, which is implied by the orientation with a decorated reduction ordering). Let $\phi : g^{:S_g} \rightarrow d^{:S_d}$ be in R. Then:

$$(t[\sigma'(l^{:S_l})] \cong_d t \rightarrowtail^{\sigma',l^{:S_l}\rightarrow r^{:S_r}}_{R} t' \cong_d t[\sigma'(r^{:S_r})] \cong_d t[\sigma'(r^{:S_r}[\sigma(g^{:S_g})])])$$
$$\Longrightarrow$$
$$(t \rightarrowtail^{\sigma',l^{:S_l}\rightarrow r'^{:S_{r'}}}_{R'} t'' \leftarrowtail^{\sigma'\circ\sigma,\phi}_{R'} t')$$

where – $r'^{:S_{r'}} \cong_d r^{:S_r}[\sigma(d^{:S_d})]$ and
– $t'' \cong_d t[\sigma'(r^{:S_r}[\sigma(d^{:S_d})])]$ (by Proposition 7.18).

If $\phi : (g^{:s} \rightarrow g^{:s\cup S_g} \text{ if } S_g \not\subseteq s) \in D$, we get:

$$(t[\sigma'(l^{:S_l})] \cong_d t \rightarrowtail^{\sigma',l^{:S_l}\rightarrow r^{:S_r}}_{R} t' \cong_d t[\sigma'(r^{:S_r})] \cong_d t[\sigma'(r^{:S_r}[\sigma(g^{:U})])])$$
$$\Longrightarrow$$
$$(t \rightarrowtail^{\tau,l^{:S_l}\rightarrow r'^{:S_{r'}}}_{R'} t'' \leftarrowtail^{\sigma'\circ\sigma,\phi}_{D'} t')$$

where – $t'' \cong_d t[\sigma'(r^{:S_r}[\sigma(g^{:U})^{:U' \cup S_g}])]$ by Proposition 7.18,
– $U' = Deco(\sigma'(\sigma(g^{:U})))$ and
– $r'^{:S_{r'}} =_d r^{:S_r}[\sigma(g^{:U})^{:U' \cup S_g}]$.

7. Compose_D_deco:
Let $\phi : (l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s)$, $\phi' : (p^{:s} \to p^{:s \cup S}$ if $S \not\subseteq s)$ and $\phi'' : (p'^{:s} \to p'^{:s \cup S'}$ if $S' \not\subseteq s)$. As in the **Simplify** case we can conclude that $\sigma, \sigma', \sigma' \circ \sigma$ (for the implicit definition of σ' see below) are decorated substitutions.
Case $\omega \neq \Lambda$, i.e. $S = S'$ and $p'^{:S'} =_d p^{:S}[\sigma(l^{:U})^{:U \cup S_l}]_\omega$:

$$(t[\sigma'(p^{:U_1}[\sigma(l^{:U_2})]_\omega)]_\nu \cong_d t \rightarrowtail_D^{\sigma',\phi'} t' \cong_d t[\sigma'(p^{:U_1}[\sigma(l^{:U_2})]_\omega)^{:U_1 \cup S}]_\nu)$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t'' \rightarrowtail_{D'}^{\tau,\phi''} t''' \leftarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t')$$

where – $t'' \cong_d t[\sigma'(p^{:U_1})]_\nu[\sigma' \circ \sigma(l^{:U_2})^{:U_2' \cup S_l}]_{\nu,\omega}$,
– $t''' \cong_d t[\sigma'(p^{:U_1})^{:U_1 \cup S}]_\nu[\sigma' \circ \sigma(l^{:U_2})^{:U_2' \cup S_l}]_{\nu,\omega}$ by Proposition 7.18,
– U_2' is $Deco(\sigma'(\sigma(l^{:U_2})))$ and
– τ is $\sigma' \circ \sigma$.
Case $\omega = \Lambda$, i.e. $S' = S \cup S_l$ and $p^{:\emptyset} =_d p'^{:\emptyset}$:

$$(t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)] \cong_d t[\sigma'(p^{:U})] \cong_d t \rightarrowtail_D^{\sigma',\phi'} t' \cong_d t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)^{:U' \cup S}])$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{D'}^{\sigma',\phi''} t'' \leftarrowtail_{D'}^{\tau,\phi} t')$$

where – $t'' \cong_d t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)^{:U' \cup S \cup S_l}]$ (by Proposition 7.18),
– $U' = Deco(\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda))$ and
– τ is $\sigma' \circ \sigma$.

8. Compose_R_deco:
Like with **Simplify**, we can argue that σ, σ' and $\sigma \circ \sigma'$ are decorated substitutions. Let $\phi : l^{:S_l} \to r^{:S_r}$, $\phi' : (p^{:s} \to p^{:s \cup S}$ if $S \not\subseteq s)$ and $\phi'' : (p'^{:s} \to p'^{:s \cup S}$ if $S \not\subseteq s)$, then:

$$(t[\sigma'(p^{:U}[\sigma(l^{:S_l})])] \cong_d t[\sigma'(p^{:U})] \cong_d t \rightarrowtail_D^{\sigma',\phi'} t' \cong_d t[\sigma'(p^{:U}[\sigma(l^{:S_l})])^{:U \cup S}])$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{R'}^{\sigma' \circ \sigma,\phi} t'' \rightarrowtail_{D'}^{\sigma',\phi''} t''' \leftarrowtail_{R'}^{\sigma' \circ \sigma,\phi} t')$$

where – $t'' \cong_d t[\sigma'(p^{:U}[\sigma(r^{:S_r})])]$ and
– $t''' \cong_d t[\sigma'(p^{:U}[\sigma(r^{:S_r})])^{:U \cup S}]$ (by Proposition 7.18).

9. Subsume_deco:
As in the **Simplify** case we may conclude that $\sigma' \circ \sigma$ and τ in the following are decorated substitutions. Thus:

$$(t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)] \cong_d t[\sigma'(p^{:U})] \cong_d t \rightarrowtail_D^{\sigma',\phi'} t' \cong_d t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)^{:U \cup S}])$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t'' \overset{0,1}{\leftarrowtail}_{D'}^{\tau,\phi} t')$$

where – ϕ is $l^{:s} \to l^{:s \cup S_l}$ if $S_l \not\subseteq s$,
– ϕ' is $p^{:s} \to p^{:s \cup S}$ if $S \not\subseteq s$,
– $t'' \cong_d t[\sigma'(p^{:U}[\sigma(l^{:U})]_\Lambda)^{:U \cup S_l}]$ (by Proposition 7.18) and
– τ is $\sigma' \circ \sigma$.

10. Collapse:
As in the **Simplify** case we may conclude that $\sigma' \circ \sigma$ is a decorated substitution. Furthermore the condition of **Collapse** prevents us from collapsing a rule with itself and therefore we have $\phi \in D' \cup R'$.
Let $\phi : g^{:S_g} \to d^{:S_d}$ be in R. We get:

$$(t[\sigma'(l^{:S_l}[\sigma(g^{:S_g})])] \cong_d t[\sigma'(l^{:S_l})] \cong_d t \rightarrowtail_R^{\sigma', l^{:S_l} \to r^{:S_r}} t' \cong_d t[\sigma'(r^{:S_r})])$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{R'}^{\sigma' \circ \sigma, \phi} t'' \longleftrightarrow_{E'}^{\sigma', l'^{:S_{l'}} = r^{:S_r}} t')$$

where – $l'^{:S_{l'}} \cong_d l^{:S_l}[\sigma(d^{:S_d})]$ and
– $t'' \cong_d t[\sigma'(l^{:S_l}[\sigma(d^{:S_d})])]$ (by Proposition 7.18).
Else $\phi : (g^{:s} \to g^{:s \cup S_g}$ if $S_g \not\subseteq s) \in D$ and we have:

$$(t[\sigma'(l^{:S_l})] \cong_d t \rightarrowtail_R^{\sigma', l^{:S_l} \to r^{:S_r}} t' \cong_d t[\sigma'(r^{:S_r})])$$
$$\Longrightarrow$$
$$(t \rightarrowtail_{D'}^{\sigma' \circ \sigma, \phi} t'' \longleftrightarrow_{E'}^{\sigma' \circ \sigma, l'^{:S_{l'}} = r^{:S_r}} t')$$

where – $t'' \cong_d t[\sigma'(l^{:S_l}[\sigma(g^{:U})^{:U' \cup S_g}])]$ (by Proposition 7.18),
– $U' = Deco(\sigma'(\sigma(g^{:U})))$ and
– $l'^{:S_{l'}} \cong_d l^{:S_l}[\sigma(g^{:U})^{:U' \cup S_g}]$.

□

10.3 Proof Reduction and Reflection

Let us now extend the set of proof reduction rules $\Longrightarrow$ introduced in the last section. The last case to consider are peaks which are implicitly reducible in any decorated presentation, i.e. peaks without critical pairs. Theorem 9.13 allows reducing such peaks by proof reduction rules of the form:

$$t'^{:S'} \hookleftarrow_{D \cup R}^{\phi'} t^{:S} \rightarrowtail_{D \cup R}^{\phi''} t''^{:S''} \Longrightarrow t'^{:S'} \overset{*}{\rightarrowtail}_{D \cup R}^{\phi''} t_0^{:S_0} \overset{*}{\hookleftarrow}_{D \cup R}^{\phi'} t''^{:S''},$$

called Peak without overlap and Peak with variable overlap according to the current peak type. Note that t, t', t'' denote any decorated terms in this rule. The next step is to prove that $\Longrightarrow$ is well-founded.

Lemma 10.4 *The proof reduction relation $\Longrightarrow$ is well-founded.*

Proof: Define the complexity measure of an elementary proof steps by:

$$\begin{aligned} c(t^{:S_1} \rightarrowtail_D^{\phi} t^{:S_2}) &= (\{t^{:S_1}\}, \phi, t^{:S_2}), \\ c(t^{:S_1} \longleftrightarrow_E^{\phi} t'^{:S_2}) &= (\{t^{:S_1}, t'^{S_2}\}, \phi, -), \\ c(t^{:S_1} \rightarrowtail_R^{\phi} t'^{:S_2}) &= (\{t^{:S_1}\}, \phi, t'^{:S_2}). \end{aligned}$$

By convention, the complexity of the empty proof Φ is $c(\Phi) = \emptyset$. Complexities of elementary proof steps are compared using the lexicographic combination denoted $>_{ec}$ of the multiset extension of $>_d$ on decorated terms, $\gg$ on formulas and again $>_d$ on decorated terms. Since $>_d$ and $\gg$ are well-founded, so is $>_{ec}$.

The complexity of a non-elementary proof is the multiset of the complexities of its proof steps. Complexities of non-elementary proofs are compared using the multiset extension $>_c$ of $>_{ec}$, which is also well-founded. Remark that $t \cong_d u$, $t' \cong_d u'$ and $t <_d t'$ implies $u <_d u'$, by definition 7.21 of decorated reduction orderings. Therefore, we can work with a representative of the equivalence class modulo $\cong_d$ of a term occurring in a proof.

- Orient_(N)SD:
  Let $t \cong_d t[\sigma(p^{:S})]$ and $t' \cong_d t[\sigma(q^{:S'})]$.
  $c(t \longleftrightarrow_E^{\sigma, p^{:S}=q^{:S'}} t') = \{(\{t,t'\}, p^{:S} = q^{:S'}, -)\} >_c$
  $c(t[\sigma(p^{:S})] \rightarrowtail_{R'}^{\sigma', p^{:S}\to q^{:S\cup S'}} t[\sigma(q^{:S'})^{:U}] \overset{0,1}{\hookleftarrow}_{D'}^{\sigma,\phi} t[\sigma(q^{:S'})]) =$
  $\{(\{t\}, p^{:S} \to q^{:S\cup S'}, t[\sigma(q^{:S'})^{:U}])\} \cup H,$
  where H is $\emptyset$ or $\{(\{t'\}, \phi, t[\sigma(q^{:S'})^{:U}])\}$ and ϕ is $q^{:s} \to q^{:s\cup S\setminus S'}$ if $S \setminus S' \not\subseteq s$,
  since $\{t,t'\} >_d^{mult} \{t\}, \{t'\}$.

- Deduce:
  $c(t' \leftarrowtail_{R\cup D}^{\sigma,\phi} t \rightarrowtail_R^{\sigma',\phi'} t'') = \{(\{t\}, \phi, t'), (\{t\}, \phi', t'')\} >_c$
  $c(t' \longleftrightarrow_{E'}^{\nu,\tau, p^{:S}=q^{:S'}} t'') = \{(\{t',t''\}, p^{:S} = q^{:S'}, -)\}$
  just by comparing the first components, since $t >_d t'$ and $t >_d t''$.

- Deduce_deco:
  $c(t' \leftarrowtail_{D\cup R}^{\sigma',\phi'} t \rightarrowtail_D^{\sigma,\phi} t'') = \{(\{t\}, \phi', t'), (\{t\}, \phi, t'')\} >_c$
  $c(t' \rightarrowtail_{D'}^{\tau,\phi''} t'_0 \overset{0,1}{\hookleftarrow}_{D'\cup R'}^{\sigma',\phi'} t'') = \{(\{t'\}, \phi', t'_0)\} \cup H,$
  where H is $\emptyset$ or $\{(\{t''\}, \phi', t'_0)\}$ and ϕ'' is $(p^{:s} \to p^{:s\cup S}$ if $S \not\subseteq s)$
  just by comparing the first components, since $t >_d t'$ and $t >_d t''$.

- Simplify:
  case $\phi \in R$:
  $c(t \longleftrightarrow_E^{\sigma', p^{:S}=q^{:S'}} t') = \{(\{t,t'\}, p^{:S} = q^{:S'}, -)\} >_c$
  $c(t \rightarrowtail_{R'}^{\sigma'\circ\sigma,\phi} t'' \longleftrightarrow_{E'}^{\sigma', p''^{:S''}=q^{:S'}} t') = \{(\{t\}, \phi, t''), (\{t'',t'\}, p''^{:S''} = q^{:S'}, -)\},$
  where $t \cong_d t[\sigma'(p^{:S}[\sigma(g^{:S_g})])]$, $t' \cong_d t[\sigma'(q^{:S'})]$ and $t'' \cong_d t[\sigma'(p^{:S}[\sigma(d^{:S_d})])]$,
  since $\{t,t'\} >_d^{mult} \{t\}$ and $\{t,t'\} >_d^{mult} \{t'',t'\}$.

  case $\phi \in D$:
  $c(t \longleftrightarrow_E^{\sigma', p^{:S}=q^{:S'}} t') = \{(\{t,t'\}, p^{:S} = q^{:S'}, -)\} >_c$
  $c(t \rightarrowtail_{D'}^{\sigma'\circ\sigma,\phi} t'' \longleftarrow_{E'}^{\tau, p''^{:S''}=q^{:S'}} t') = \{(\{t\}, \phi, t''), (\{t',t''\}, p''^{:S''} = q^{:S'}, -)\},$
  where $t \cong_d t[\sigma'(p^{:S}[\sigma(l^{:U})]_\omega)]_\nu$, $t' \cong_d t[\sigma'(q^{:S'})]_\nu$ and $t'' \cong_d t[\sigma'(p^{:S})]_\nu[\sigma' \circ \sigma(l^{:U})^{:U'\cup S_l}]_{\nu.\omega}$.
  Clearly, $\{t,t'\} >_d^{mult} \{t\}$ and $\{t,t'\} >_d^{mult} \{t',t''\}$ gives the reduction of the complexity measure.

- Delete:
  $c(t \longleftrightarrow_E^{p^{:S}=p^{:S}} t) = \{(\{t,t\}, p = p, -)\} >_c c(\Phi) = \emptyset$
  since $\{t,t\} >_d^{mult} \emptyset$.

- Compose:
  case $\phi \in R$:
  $c(t \rightarrowtail_R^{\sigma', l^{:S_l}\to r^{:S_r}} t') = \{(\{t\}, l^{:S_l} \to r^{:S_r}, t')\} >_c$
  $c(t \rightarrowtail_{R'}^{\sigma', l^{:S_l}\to r'^{:S_{r'}}} t'' \leftarrowtail_{R'}^{\sigma'\circ\sigma,\phi} t') = \{(\{t\}, l^{:S_l} \to r'^{:S_{r'}}, t''), (\{t'\}, \phi, t'')\},$
  where $t \cong_d t[\sigma'(l^{:S_l})]$, $t' \cong_d t[\sigma'(r^{:S_r}[\sigma(g^{:S_g})])]$ and $t'' =_d t[\sigma'(r^{:S_r}[\sigma(d^{:S_d})])]$,
  since $r^{:S_r} >_d r'^{:S_{r'}}$ – this implies $(l^{:S_l} \to r^{:S_r}) \gg (l^{:S_l} \to r'^{:S_{r'}})$ – and $t >_d t'$.

  case $\phi \in D$:
  $c(t \rightarrowtail_R^{\sigma', l^{:S_l}\to r^{:S_r}} t') = \{(\{t\}, l^{:S_l} \to r^{:S_r}, t')\} >_c$

$c(t \rightarrowtail_{R'}^{\tau,l:S_l \to r':S_{r'}} t'' \leftarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t') = \{(\{t\}, l:S_l \to r':S_{r'}, t''), (\{t'\}, \phi, t'')\}$,
where $t \cong_d t[\sigma'(l:S_l)]$, $t' \cong_d t[\sigma'(r:S_r[\sigma(g:U)])]$ and $t'' \cong_d t[\sigma'(r:S_r[\sigma(g:U):U \cup S_g])]$,
since $r:S_r >_d r':S_{r'}$ - this implies once more $(l:S_l \to r:S_r) \gg (l:S_l \to r':S_{r'})$ - and $t >_d t'$.

- Compose_D_deco:
  case $\omega \neq \Lambda$:
  $c(t \rightarrowtail_D^{\sigma',\phi'} t') = \{(\{t\}, \phi', t')\} >_c$
  $c(t \rightarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t'' \rightarrowtail_{D'}^{\tau,\phi''} t''' \leftarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t') = \{(\{t\}, \phi, t''), (\{t''\}, \phi'', t'''), (\{t'\}, \phi, t''')\}$,
  where $t \cong_d t[\sigma'(p:U_1[\sigma(l:U_2)]_\omega)]_\nu$, $t' \cong_d t[\sigma'(p:U_1[\sigma(l:U_2)]_\omega):U_1 \cup S]_\nu$,
  $t'' \cong_d t[\sigma'(p:U_1)]_\nu[\sigma' \circ \sigma(l:U_2):U_2' \cup S_l]_{\nu,\omega}$ and $t''' \cong_d t[\sigma'(p:U_1):U_1 \cup S]_\nu[\sigma' \circ \sigma(l:U_2):U_2' \cup S_l]_{\nu,\omega}$,
  since $l:S_\omega \not\trianglelefteq p:\emptyset$, where $Deco(p:\emptyset|_\omega) = S_\omega$, therefore $\phi' \gg \phi$, and trivially $t >_d t'$ resp. $t >_d t''$.
  case $\omega = \Lambda$:
  $c(t \rightarrowtail_D^{\sigma',\phi'} t') = \{(\{t\}, \phi', t')\} >_c c(t \rightarrowtail_{D'}^{\sigma',\phi''} t'' \leftarrowtail_{D'}^{\tau,\phi} t') = \{(\{t\}, \phi'', t''), (\{t'\}, \phi, t'')\}$,
  where $t \cong_d t[\sigma'(p:U[\sigma(l:U)]_\Lambda)]$, $t' \cong_d t[\sigma'(p:U[\sigma(l:U)]_\Lambda):U' \cup S]$
  and $t'' \cong_d t[\sigma'(p:U[\sigma(l:U)]_\Lambda):U' \cup S \cup S_l]$,
  since either $l:\emptyset \not\trianglelefteq p:\emptyset$ or $S \subset S' = S \cup S_l$, therefore $\phi' \gg \phi''$, and trivially $t >_d t'$.
- Compose_R_deco:
  $c(t \rightarrowtail_D^{\sigma',\phi'} t') = \{(\{t\}, \phi', t')\} >_c$
  $c(t \rightarrowtail_{R'}^{\sigma' \circ \sigma,\phi} t'' \rightarrowtail_{D'}^{\sigma',\phi''} t''' \leftarrowtail_{R'}^{\sigma' \circ \sigma,\phi} t') = \{(\{t\}, \phi, t''), (\{t''\}, \phi'', t'''), (\{t'\}, \phi, t''')\}$,
  where
  $t \cong_d t[\sigma'(p:U[\sigma(l:S_l)])]$, $t' \cong_d t[\sigma'(p:U[\sigma(l:S_l)]):U \cup S]$, $t'' \cong_d t[\sigma'(p:U[\sigma(r:S_r)])]$
  and $t''' \cong_d t[\sigma'(p:U[\sigma(r:S_r)]):U \cup S]$,
  because of $\phi' \gg \phi$ (since $\omega \neq \Lambda$ and therefore $l:S_l$ must be strictly embedded in $p:S$), and finally $t >_d t'', t'$.
- Subsume_deco:
  $c(t \rightarrowtail_D^{\sigma',\phi'} t') = \{(\{t\}, \phi', t')\} >_c c(t \rightarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t'' \leftarrowtail_{D'}^{\tau,\phi} t') = \{(\{t\}, \phi, t''), (\{t'\}, \phi, t'')\}$
  where $t \cong_d t:T[\sigma'(p:U[\sigma(l:U)]_\Lambda)]$, $t' \cong_d t:T[\sigma'(p:U[\sigma(l:U)]_\Lambda):U \cup S]$ and $t'' \cong_d t:T[\sigma'(p:U[\sigma(l:U)]_\Lambda):U \cup S_l]$,
  because of $\phi' \gg \phi$ (since $\phi' \neq \phi$ we have either $p:\emptyset$ as strict instance of $p':\emptyset$ - then we are done - or the two terms are equal modulo variable renaming implying $S \subset S'$ and therefore $p:S >_d p':S'$). Finally, $t >_d t'$, since $U \not\subseteq S$.

- Collapse:
  case $\phi \in R$:
  $c(t \rightarrowtail_R^{\sigma',l:S_l \to r:S_r} t') = \{(\{t\}, l:S_l \to r:S_r, t')\} >_c$
  $c(t \rightarrowtail_{R'}^{\sigma' \circ \sigma,\phi} t'' \leftarrowtail\!\!\rightarrowtail_{E'}^{\sigma',l':S_{l'}=r:S_r} t') = \{(\{t\}, \phi, t''), (\{t'', t'\}, l':S_{l'} = r:S_r, -)\}$
  where $t \cong_d t[\sigma'(l:S_l[\sigma(g:S_g)])]$, $t' \cong_d t[\sigma'(r:S_r)]$ and $t'' \cong_d t[\sigma'(l:S_l[\sigma(d:S_d)])]$,
  since $(l:S_l \to r:S_r) \gg \phi$ and $t >_d t', t''$.
  case $\phi \in D$:
  $c(t \rightarrowtail_R^{\sigma',l:S_l \to r:S_r} t') = \{(\{t\}, l:S_l \to r:S_r, t')\} >_c$
  $c(t \rightarrowtail_{D'}^{\sigma' \circ \sigma,\phi} t'' \leftarrowtail\!\!\rightarrowtail_{E'}^{\tau,l':S_{l'}=r:S_r} t') = \{(\{t\}, \phi, t''), (\{t'', t'\}, l':S_{l'} = r:S_r, -)\}$,
  where $t \cong_d t[\sigma'(l:S_l)]$, $t' \cong_d t[\sigma'(r:S_r)]$ and $t'' \cong_d t[\sigma'(l:S_l[\sigma(g:U):U' \cup S_g])]$,
  because of similar reasons as in the case of $\phi \in R$.
- Peak without overlap:
  $c(t' \leftarrowtail_{D \cup R}^{\phi} t \rightarrowtail_{D \cup R}^{\phi'} t'') = \{(\{t\}, \phi, t'), (\{t\}, \phi', t'')\} >_c$

$c(t' \rightarrowtail^{\phi'}_{D \cup R} t_1 \leftarrowtail^{\phi}_{D \cup R} t'') = \{(\{t'\}, \phi', t_1), (\{t''\}, \phi, t_1)\}$
just by comparing the first components: $t >_d t'$ and $t >_d t''$.

- Peak with variable overlap:
  $c(t' \leftarrowtail^{\phi}_{D \cup R} t \rightarrowtail^{\phi'}_{D \cup R} t'') >_c c(t' \overset{*}{\rightarrowtail}_{D \cup R} t_1 \overset{*}{\leftarrowtail}_{D \cup R} t'')$
  again just by comparing the first components of each elementary step appearing in these proofs.

□

The exact correspondence between the proof reduction $\Longrightarrow$ and the derivation $\vdash_{OSC}$ is stated by the following results.

These proof reduction rules must *reflect* the rules of $\mathcal{OSC}$ in the following sense: at each step $\vdash_{OSC}$, a given proof either does not change or is transformed into another one by $\Longrightarrow$.

Definition 10.5 *[Bac91] The proof reduction* $\Longrightarrow$ reflects $\vdash_{OSC}$ *if whenever* $(D_i, E_i, R_i) \vdash_{OSC} (D_{i+1}, E_{i+1}, R_{i+1})$ *and* P *is a proof in* (D_i, E_i, R_i)*, then there is a proof* P' *in* $(D_{i+1}, E_{i+1}, R_{i+1})$ *such that* $P \overset{*}{\Longrightarrow} P'$.

Because the $\Longrightarrow$-rules have been built from the rules of $\mathcal{OSC}$, it is easy to verify that:

Proposition 10.6 $\Longrightarrow$ *reflects* $\vdash_{OSC}$.

10.4 Fairness

The fairness hypothesis states that any proof reducible by $\Longrightarrow$ will eventually be reduced. In other words, no reducible proof is forgotten. Fairness specifies under which conditions a control for applying rules of $\mathcal{OSC}$ is correct. Fairness is again defined relatively to a subset $\mathcal{T}$ of valid terms.

Definition 10.7 *Let* $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$*. A derivation* $(D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ *is* $\mathcal{T}$-fair *if whenever* Ψ *is a proof in* (D_i, E_i, R_i)*, that uses only terms in* $\mathcal{T}$ *and is reducible by* $\Longrightarrow$*, then there is a proof* Ψ' *in* (D_j, E_j, R_j) *at some step* $j \geq i$ *such that* $\Psi \overset{+}{\Longrightarrow} \Psi'$.

When $\mathcal{T} = \mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, we drop the $\mathcal{T}$ prefix, since the notion is then obviously equivalent to classical fairness. Let us define:

$$D_* = \bigcup_{i \geq 0} D_i \qquad E_* = \bigcup_{i \geq 0} E_i \quad \text{and} \quad R_* = \bigcup_{i \geq 0} R_i$$
$$D_\infty = \bigcup_{i \geq 0} \bigcap_{j > i} D_j \quad E_\infty = \bigcup_{i \geq 0} \bigcap_{j > i} E_j \quad \text{and} \quad R_\infty = \bigcup_{i \geq 0} \bigcap_{j > i} R_j.$$

A sufficient condition to satisfy the fairness hypothesis can be given:

Proposition 10.8 *Let* $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ *and* $\mathbf{UNIF_d}$ *a strict_subterm_set*$(\mathcal{T})$*-complete unification algorithm. A derivation* $(D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ *using* $\mathbf{UNIF_d}$ *for the calculation of critical pairs is* $\mathcal{T}$*-fair if* E_∞ *is empty and all critical pairs of* $D_\infty \cup R_\infty$ *are in* $D_* \cup E_*$.

Proof: We have to prove that if Ψ is a proof in (D_i, E_i, R_i) that uses only terms in $\mathcal{T}$ and is reducible by $\Longrightarrow$, then there is Ψ' in (D_j, E_j, R_j) at some step $j \geq i$ such that $\Psi \overset{+}{\Longrightarrow} \Psi'$.

If one of the transformation rules Orient_SD/NSD, Simplify, Delete, Compose, Compose_D_deco / _R_deco, Subsume_deco, or Collapse applies to Ψ, then $(D_i, E_i, R_i) \neq (D_\infty, E_\infty, R_\infty)$. So one of the rules of $\mathcal{OSC}$ applies.

If the transformation rule Deduce or Deduce_deco applies with a non-persisting rule then for some $j \geq i$, $(D_i, E_i, R_i) \vdash \ldots \vdash (D_j, E_j, R_j)$ where (D_j, E_j, R_j) does not contain this rule any

Extend	$\dfrac{D, E \cup \{p^{:S \cup \{B\}} = (x :: A)^{:S'}\}, R}{\bot, \bot, \bot}$ if $x \in \mathcal{X}_\Diamond$ and $x :: A \in \mathcal{P}$ and not $A \leq_S^{syn} B$

Figure 9: Test for Equivalence of $\leq_S^{syn}$ and $\leq_S^{sem}$.

more. If the transformation rule Deduce or Deduce_deco applies with persisting rules, then by hypothesis there exists $k > i$ such that the computed critical pair is in $D_k \cup E_k$, since all terms in Ψ are in $\mathcal{T}$ and $\mathbf{UNIF_d}$ is complete for the unification problems in $CP(R,R)_{|_\mathcal{T}}$, $CP(R,D)_{|_\mathcal{T}}$, $CP(D,D)_{|_\mathcal{T}}$ and $CP(D,R)_{|_\mathcal{T}}$ as a consequence of Lemma 9.8. Remark, that since $E_\infty = \emptyset$ by hypothesis, then for some $j > k > i$, $(D_i, E_i, R_i) \vdash \ldots \vdash (D_j, E_j, R_j)$ where E_j does not contain the critical pair any more.

For all these cases, since $\Longrightarrow$ reflects $\vdash$, $P \stackrel{+}{\Longrightarrow} P'$.

If either Peak without overlap, or Peak with variable overlap applies, then by Theorem 9.13, P contains a peak that can be replaced by a rewrite proof $t' \stackrel{*}{\rightarrow} u \stackrel{*}{\leftarrow} t''$. Then $j = i$ and $P \stackrel{+}{\Longrightarrow} P'$. □

For a fair derivation, the resulting decorated presentation satisfies the property that any equational proof has a normal form which is a rewrite proof.

We then get the main result of this section:

Theorem 10.9 *Let $\mathcal{P}_0$ be sort inheriting and $(D_0, E_0, R_0) \vdash_{OSC} (D_1, E_1, R_1) \vdash_{OSC} \ldots$ be a fair derivation. Then $\mathcal{P}_\infty$ is terminating, Church-Rosser, type complete and existentially complete on $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$. Moreover $Th(\mathcal{P}_0) = Th(\mathcal{P}_\infty)$.*

Proof: The termination property of $D_\infty \cup R_\infty$ is obvious since the test is incrementally processed for each rule added in $D_* \cup R_*$, thus in $D_\infty \cup R_\infty$.

Then the derivation is $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-fair and any proof $t^{:\downarrow\emptyset} \stackrel{*}{\leftrightarrow}_{D_0 \cup E_0 \cup R_0} t'^{:S'}$ has a rewrite proof in $\mathcal{P}_\infty$.

This guarantees the Church-Rosser property, as well as type and existential completeness. Finally $Th(\mathcal{P}_0) = Th(\mathcal{P}_\infty)$ is a consequence of Proposition 10.3. □

It should be noticed that the presented rules for completion do not include rules for simplifying at the top occurrence decoration rules with decorated rewrite rules. This is solved by decoration critical pairs that add the simplified versions of the decoration rules without deleting their old versions. So a lot of decoration rules may be generated during a fair derivation. However, including more simplification for decoration rules is yet an unsolved point.

10.5 Changing the Subsort Relation

The unification algorithm needed for the completion process assumes, of course, the subsort relation to stay static, but this may not be the case (see example 4.3). The test for equality of $\leq_S^{syn}$ and $\leq_S^{sem}$ can be done using the **Extend** rule shown in figure 9, which can be added to OSC.

The rule **Extend** is semi-complete when it is used in OSC together with a fair strategy.

Proposition 10.10 *Let $\leq_S^{syn}$ be a syntactic sort ordering used for the completion of $\mathcal{P}$, a sort inheriting presentation w.r.t. $\leq_S^{syn}$. Let furthermore $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using OSC with a fair strategy if $\mathcal{P}_\infty \neq (\bot, \bot, \bot)$.*

Extend *applies iff $\leq_S^{syn} \neq \leq_S^{sem}$.*

Proof: First of all, if **Extend** is applicable, then clearly $\leq_{\mathcal{S}}^{syn} \neq \leq_{\mathcal{S}}^{sem}$. Now, suppose $\leq_{\mathcal{S}}^{syn} \neq \leq_{\mathcal{S}}^{sem}$ and **Extend** does not apply. Hence, there are sorts A, B, s.t. $A \leq_{\mathcal{S}}^{sem} B$, but not $A \leq_{\mathcal{S}}^{syn} B$. Therefore Lemma 8.1 implies the existence of a corresponding proof $\Psi : x^{:\{A\}} \overset{*}{\leftrightarrow}_{\mathcal{P}_0} t^{:S \cup \{B'\}}$ with $B' \leq_{\mathcal{S}}^{syn} B$ for a variable x with $(x :: A) \in \mathcal{P}$.

From fairness and termination of $\Longrightarrow$ (see proposition 10.4), it follows now that there exists a rewrite proof form Ψ_k of Ψ after a finite number of steps k, since any peak must be reduced after a finite number of steps yielding a strictly smaller proof. Let $t'^{:S'}$ be the normal form of $x^{:\emptyset} =_d x^{:\{A\}}$ in Ψ_k.

Assume $t' \notin \mathcal{X}_0$. Clearly, some $A' \leq_{\mathcal{S}}^{syn} A$ must be in S', since decoration and decorated rewriting can only increase the top decoration. Therefore, $\sigma = \{x^{:\{A\}} \mapsto t'^{:S'}\}$ is a decorated substitution. If $x \in \mathcal{V}ar(t'^{:S'})$, then $\sigma(x^{:\{A\}})$ is a strict subterm of $\sigma(t'^{:S'})$. Otherwise, if $x \notin \mathcal{V}ar(t'^{:S'})$, then $\sigma(x^{:\{A\}}) \cong_d \sigma(t'^{:S'})$. In both cases, we get a contradiction to the well-foundedness of $<_d$, since $\sigma(x^{:\{A\}})$ rewrites to another term $t'^{:S'}$ that contains $\sigma(x^{:\{A\}})$. Hence, $t' \notin \mathcal{X}_0$ is impossible, not even for intermediate terms in Ψ_k.

Moreover, $t' = x$ must hold, because of $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ for any decorated rewrite rule $l \to r$ by definition. So, we can be sure that $\Psi_k : x^{:\emptyset} \overset{*}{\rightarrowtail}_{R_k} x^{:T \cup \{B'\}}$ and any $(\phi_i : l_i^{:S_i} \to r_i^{:S_i'}) \in R_k$ used in Ψ_k must satisfy $(l_i)_{nd} = (r_i)_{nd} = x$ and $S_i \subset S_i'$. Remark that decoration rewrite rules cannot be used in Ψ_k, because variable decoration rules are inapplicable due to an unsatisfiable condition.

Consequently, one of the S_i' must contain B'. But this is in contradiction to the conditions for orientation rules in $\mathcal{OSC}$, i.e. the equation from which ϕ' stems cannot have been oriented and Ψ_k cannot exist. So, **Extend** must be applicable.

□

In the case where **Extend** applies, we just proved that $\leq_{\mathcal{S}}^{syn} \subset \leq_{\mathcal{S}}^{sem}$ and the syntactical ordering on sorts should be extended before a new attempt of completion. But several situations may then occur, as in the following examples, where different kinds of extensions to the syntactical ordering are illustrated.

Example 10.11 *Let $\mathcal{S} = \{A, B, C, D\}$, $\leq_1$ be the initial syntactic subsort relation and $\leq_2$ be the one where the new extensions are added. We can add in $\leq_2$ a relation between :*

- *incomparable sorts without common subsorts : If $\leq_1 = \emptyset$ and $\leq_2 = \{(A, B)\}$, then any solution of a unification problem of the form $(x :: A \cong_d^? y :: B)$ was incomplete. So we have to restart the critical pair computation.*
- *comparable sorts : If $\leq_1 = \{(A, B)\}$ and $\leq_2 = \{(A, B), (B, A)\}$, then we added a cycle. To satisfy Assumption 4.6 again, we have to replace A or B in the last presentation by a unique representative sort, in order to continue the computations. This possibly makes some decorated rewrite rules trivial. We may need to re-orient some rules. If some condition of a decoration rewrite rule becomes unsatisfiable and this rule can of course be deleted.*
- *incomparable sorts with a common subsort : If $\leq_1 = \{(C, A), (C, B), (D, A)\}$ and $\leq_2 = \{(C, A), (D, A), (C, B), (D, B)\}$, then we get $D \leq_{\mathcal{S}_0}^{syn} \langle A, B \rangle$. Consequently, the subsort relation has to be completed as in the first case and the whole completion has to be reset, too.*

11 Checking Sort Inheritance

We are now left with the problem of checking the sort inheritance property of a decorated presentation. The idea is to characterize non sort inheritance on a set of decorated terms $\mathcal{T}$ by a property of the

Detect	$\dfrac{D \cup \{(p^{:s} \to p^{:s \cup \{A\} \cup S} \text{ if } c(s)), (p'^{:s} \to p'^{:s \cup \{B\} \cup S'} \text{ if } c(s))\}, E, R}{\bot, \bot, \bot}$ if $\exists \psi \ : \ \psi(p^{:\emptyset}) = \psi(p'^{:\emptyset})$ and $\nexists C \leq A, B$

Figure 10: Sort Inheritance Test Rule.

decoration rule set: D must contain two rules whose left-hand sides are unifiable and that produce the adjunction of sorts A and B respectively without common subsort $C \leq A, B$. This characterization is possible if D is confluent on $\mathcal{T}$ and any term of $\mathcal{T}$ is reachable by D only. A corresponding test can be realized by a rule **Detect** shown in Figure 10.

Given a confluent and terminating set of decoration rules D, a *typing proof* of a term $t^{:T}$ (with $T \neq \emptyset$) is of the form $t^{:\downarrow\emptyset} \overset{*}{\leftarrow\!\!\rightarrow}_D t^{:T}$ and terms with a typing proof are simply called *typable*. Proofs over typable terms only are therefore called typable, too.

The first step is to prove that our test on decoration rules is sufficient to ensure sort inheritance on $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$, i.e. all typable terms, if D is confluent and terminating. The second step is the observation that the proofs constructed by Theorem 8.1 are typable.

This motivates the search of a proof reduction maintaining this property and thus called *typing conservative*. So the third step is to exhibit a proof transformation that preserves typability of terms. Using this transformation we then prove that any irreducible proof is also typable. We thus get sort inheritance on the set of valid decorated terms $Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$: for such a term $t^{:S}$, there exists a proof $t^{:\downarrow\emptyset} \overset{*}{\leftarrow\!\!\rightarrow}_{p_0} t'^{:S}$ with an irreducible form of the form $t^{:\downarrow\emptyset} \overset{*}{\longrightarrow} t''^{:S''} \overset{*}{\longleftarrow} t'^{:S}$ for some rewrite relation $\longrightarrow$, with $S \subseteq S''$, and $t''^{:S''}$ is reachable from $t''^{:\downarrow\emptyset}$ using a confluent system D. Then if the test if sort inheritance succeeds on the set S, then it succeeds also on S''.

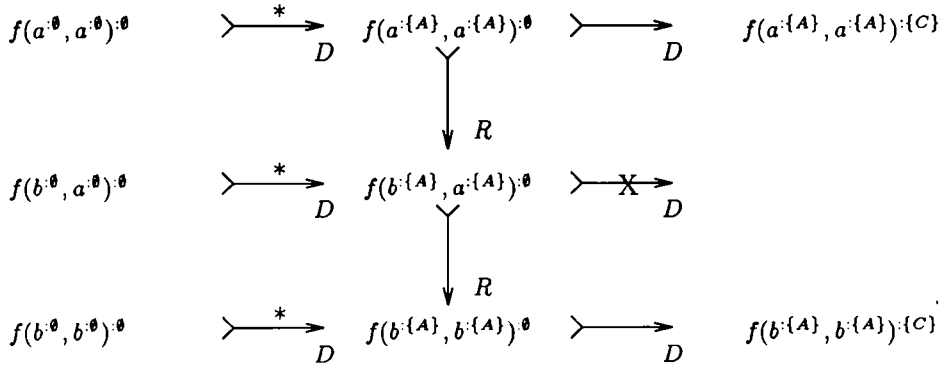
The real difficulty is to find a typing conservative proof transformation. Unfortunately, when dealing with non-linear *decoration* rules, the proof transformation $\Longrightarrow$ is too general for this propagation, since the simplification of terms can be done at arbitrary occurrences. This means that the reduction by $\rightarrowtail_R$ could destroy the typability of a term, i.e. its reachability by $\rightarrowtail_D$.

Let us give a simple example of the problem that occurs with non-linearity.

Example 11.1 *Let $\mathcal{P} = (D, \emptyset, R)$ be the following decorated presentation:*

$$\begin{aligned} D &= \{a^{:s} \to a^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s, \\ &\quad\ b^{:s} \to b^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s, \\ &\quad\ f((x :: A)^{:\{A\}}, x^{:\{A\}})^{:s} \to f(x^{:\{A\}}, x^{:\{A\}})^{:s \cup \{C\}} \text{ if } \{C\} \not\subseteq s\}, \\ R &= \{a^{:\{A\}} \to b^{:\{A\}}\} \end{aligned}$$

Then we have the following reductions:



This shows that a reduction by R may destroy the reducibility by D.

11.1 Testing Sort Inheritance On D-Closed Sets

The main goal of this section is the construction of a test for sort inheritance on the set of decorated terms $\mathcal{T}$ reachable from $\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset}$ with decoration rules only.

The notion of D-closure of a set of terms is needed to define this set.

Definition 11.2 *Let (D, E, R) be a decorated presentation. A set $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ is* D-closed *if for all $t \in \mathcal{T}$:*

1. $t^{:\downarrow\emptyset} \in \mathcal{T}$,
2. $t \overset{*}{\leftarrow\!\diamond\!\rightarrow}_D t'$ *implies* $t' \in \mathcal{T}$ *and,*
3. $t \in \mathcal{T}$ *implies* $\forall \omega \in \mathcal{O}cc(t) : t_{|\omega} \in \mathcal{T}$.

The D-closure $reach_D(\mathcal{T}^{:\downarrow\emptyset})$ of a set $\mathcal{T}^{:\downarrow\emptyset}$ of terms with empty decorations is therefore the set:

$$\{t \mid \exists t' \in \mathcal{T}^{:\downarrow\emptyset} : t' \overset{*}{\leftarrow\!\diamond\!\rightarrow}_D t'' \text{ and } \exists \omega \in \mathcal{O}cc(t'') : t =_d t''_{|\omega}\}.$$

Note that $strict\_subterm\_set(reach_D(\mathcal{T}^{:\downarrow\emptyset})) \subseteq reach_D(strict\_subterm\_set(\mathcal{T}^{:\downarrow\emptyset}))$ is an immediate consequence of the conditions 1 and 3.

Proposition 11.3 *Let $\mathcal{P} = (D, E, R)$ be a $strict\_subterm\_set(\mathcal{T})$-sort inheriting decorated presentation, such that D is confluent on $\mathcal{T} = reach_D(\mathcal{T}^{:\downarrow\emptyset})$. Then $\mathcal{P}$ is $\mathcal{T}$-sort inheriting w.r.t. $\leq_{\mathcal{S}}^{syn}$ iff for all $t \in \mathcal{T}$: $t^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_D t^{:T}$ implies that for any two sorts $A, B \in T$, there is a third sort C with $C \leq_{\mathcal{S}}^{syn} A, B$.*

Proof: $\Rightarrow$: This is obvious because of $\mathcal{T} = reach_D(\mathcal{T}^{:\downarrow\emptyset})$.

$\Leftarrow$: Since $\mathcal{P}$ is already $strict\_subterm\_set(\mathcal{T})$-sort inheriting, we only have to check top decorations. Let $t^{:T} \in \mathcal{T}$. Then $t^{:\downarrow\emptyset} \overset{*}{\leftarrow\!\diamond\!\rightarrow}_D t^{:T'}$. Since D is confluent, $t^{:\downarrow\emptyset}$ has a unique D-normal form modulo sort inheritance $t^{:T}$ and furthermore $t^{:T'} \overset{*}{\rightarrowtail}_D t^{:T}$. Since $T' \subsetneq T$, there is for any $A, B \in T$ a third sort C, s.t. $C \leq_{\mathcal{S}}^{syn} A, B$. □

In order to build a more syntactical test, a saturation process on decoration rules is designed. Let **Deduce_DD** stand for the rule **Deduce_deco** applied to decoration rules only and $\underline{Deduce\_DD}$ be the corresponding proof reduction rules in $\Longrightarrow$, including the rules for peaks.

Let furthermore $\mathcal{D}ed$ be the set of rules consisting of **Deduce_DD**, **Compose_D_deco**, **Subsume_deco**. $\underline{\mathcal{D}ed}$ is the corresponding set of proof transformation rules of $\Longrightarrow$.

Proposition 11.4 *Let $\mathcal{P} = (D, E, R)$ be a decorated presentation and $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ be D-closed. If $\mathcal{P} = (D, E, R) \vdash_{\mathcal{D}ed} \mathcal{P}' = (D', E, R)$, then $\mathcal{T}$ is also D'-closed and there exists a proof $\Psi : t \overset{*}{\leftarrow\!\diamond\!\rightarrow}_D t'$ with $terms(\Psi) \subseteq \mathcal{T}$ iff there is a proof $\Psi' : t \overset{*}{\leftarrow\!\diamond\!\rightarrow}_{D'} t'$ with $terms(\Psi') \subseteq \mathcal{T}$.*

Proof: This follows immediately from $\Longrightarrow$ and $\sim$ used in the proof of Proposition 10.3. □

Analogously to Definition 10.7, we can define *$\mathcal{T}$-fairness for decoration rules*, where the set of completion rules is restricted to $\mathcal{D}ed$ and $\Longrightarrow$ consequently to the rules in $\underline{\mathcal{D}ed}$. Clearly, the proof part of Proposition 10.8 dealing with the rules in $\underline{\mathcal{D}ed}$ proves that $\mathcal{T}$-fairness for decoration rules is implied by the condition that all critical pairs of D_∞ are in D_*. Since we did not change the rules themselves, the property of reflection of $\vdash_{\mathcal{D}ed}$ by $\underline{\mathcal{D}ed}$ is still valid.

Proposition 11.5 *Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ be a D-closed set of decorated terms and $(D, E, R) = (D_k, E_k, R_k)$ be a decorated presentation obtained by applying rules from Ded, that is $\mathcal{T}$-fair for decoration rules. Then D is $\mathcal{T}$-confluent.*

Proof: Remark that the $\mathcal{T}$-fairness implies that **Deduce_DD** is no more applicable. It follows immediately from the proof reduction rules Deduce_deco, peak without overlap, peak with variable overlap that D is confluent. □

Specific proofs with decoration rules, namely bottom-up decoration proofs, play a fundamental role in the test of sort inheritance.

Definition 11.6 *Let (D, E, R) be a decorated presentation. Then:*

$$\Psi : t_0 \rightarrowtail_D^{\omega_1} t_1 \ldots \rightarrowtail_D^{\omega_n} t_n$$

is a bottom-up decoration proof *iff for all $i, j \in [1..n]$, we have $i < j \Rightarrow \omega_i \not\prec \omega_j$.*

Proposition 11.7 *Let $\mathcal{T} = reach_D(T^{:\downarrow\emptyset})$ and D be $\mathcal{T}$-confluent. Then there exists for any decoration proof:*

$$\Psi : t_0{}^{:S_0} \rightarrowtail_D^{\omega_1} t_1{}^{:S_1} \ldots \rightarrowtail_D^{\omega_n} t_n{}^{:S_n}$$

where $t_i{}^{:S_i} \in \mathcal{T}$ for all $i \in [0..n]$ and $t_n{}^{:S_n}$ is irreducible in D, a bottom-up decoration rewrite proof for $t_0{}^{:S_0} \xrightarrow{*}_D t_n{}^{:S_n}$.

Proof: First, we need a partial proof ordering respecting the occurrence of rule application, such that bottom-up right-to-left decoration enrichment is less complex than any other strategy.
Let $\Psi = (t_i{}^{:S_i} \rightarrowtail_D^{\omega_i,\phi_i} t_{i+1}{}^{:S_{i+1}})_{i\in[1..n]}$, $\Psi' = (t'_i{}^{:S'_i} \rightarrowtail_D^{\omega'_i,\phi'_i} t'_{i+1}{}^{:S'_{i+1}})_{i\in[1..m]}$ be two decoration rewriting proofs. Then:

$$\Psi <_o \Psi' \text{ iff } \exists k \in [1 \ldots min(m,n)] : \forall i \in [1..k-1] : \omega'_i \preceq_{lex} \omega_i \text{ and } \omega'_k \prec_{lex} \omega_k.$$

W.l.o.g. we can consider a minimal proof of $t_0 \xrightarrow{*}_D t_n$ w.r.t. the proof ordering $<_o$. If the proof consists of a single rule application, we are trivially done.

Otherwise, let us assume that we have:

$$\begin{array}{ll} & t_m{}^{:S_m}[\sigma_m(u_m{}^{:U_m})]_{\omega_m}[\sigma_{m+1}(u_{m+1}{}^{:U_{m+1}})]_{\omega_{m+1}} \\ \omega_m,\underline{\phi_m},\sigma_m & t_m{}^{:S_m}[\sigma_m(u_m{}^{:U_m})^{:U'_m \cup T_m})]_{\omega_m}[\sigma_{m+1}(u_{m+1}{}^{:U_{m+1}})]_{\omega_{m+1}} \\ \omega_{m+1},\underline{\phi_{m+1}},\sigma_{m+1} & t_m{}^{:S_m}[\sigma_m(u_m{}^{:U_m})^{:U'_m \cup T_m})]_{\omega_m}[\sigma_{m+1}(u_{m+1}{}^{:U_{m+1}})^{:U'_{m+1} \cup T_{m+1}}]_{\omega_{m+1}} \end{array}$$

with $\omega_m \prec \omega_{m+1}$, i.e. $\omega_{m+1} = \omega_m.\nu$. We can assume that these two rule applications follow each other, since any rewrite step at an incomparable occurrence in between could be swapped with one of the two, resulting in a smaller proof.

Obviously, we can apply ϕ_{m+1} to $t_m{}^{:S_m}[\sigma_m(u_m{}^{:U_m})]_{\omega_m}[\sigma_{m+1}(u_{m+1}{}^{:U_{m+1}})]_{\omega_{m+1}}$. The resulting proof is strictly smaller w.r.t. $<_o$ and the $\mathcal{T}$-confluence of D guarantees that $t_n{}^{:S_n}$ is still reached. Thus, there must be an equivalent bottom-up decoration proof. □

Proposition 11.8 *Let $\mathcal{T} = reach_D(T^{:\downarrow\emptyset})$, $\mathcal{P} = (D, E, R)$ a strict_subterm_set($\mathcal{T}$)-sort inheriting decorated presentation, such that D is $\mathcal{T}$-confluent. Then, the* **Detect***-rule succeeds on D iff $\mathcal{P}$ is not $\mathcal{T}$-sort inheriting.*

Proof: $\Leftarrow$: Proposition 11.3 implies, that there exists a $t^{:\downarrow\emptyset} \in \mathcal{T}$ with $t^{:\downarrow\emptyset} \xrightarrow{*}_D t^{:U\cup\{A,B\}}$, s.t. there is no third sort C with $C \leq_{\mathcal{S}}^{syn} A, B$. Let w.l.o.g. A, B be minimal w.r.t. $\leq_{\mathcal{S}}^{syn}$ in $U \cup \{A, B\}$.

Hence there must be two decorated substitutions σ, τ and two decoration rules ($\phi : p^{:s} \to p^{:s\cup T}$ if $T \not\subseteq s$) with $A \in T$, $\sigma(p) =_{nd} t^{:\downarrow\emptyset}$ and ($\phi' : p'^{:s} \to p'^{:s\cup T'}$ if $T' \not\subseteq s$) with $B \in T'$, $\tau(p') =_{nd} t^{:\downarrow\emptyset}$, s.t.:

$$t^{:\downarrow\emptyset} \xrightarrow{*}_D t_k{}^{:S_k} \xrightarrow{\Lambda,\phi,\sigma} t_k{}^{:S_k\cup T} \xrightarrow{*}_D t_l{}^{:S_l} \xrightarrow{\Lambda,\phi',\sigma'} t_l{}^{:S_l\cup T'} \xrightarrow{*}_D t^{:S\cup\{A,B\}}.$$

Note that $p, p' \notin \mathcal{X}_\Diamond$, since the only way to introduce new variable decoration rule is **Orient_NSD**. Remark that decoration critical pairs are defined for $\omega \neq \Lambda$ only. **Orient_NSD** guarantees that the condition of the variable decoration rule is unsatisfiable. Consequently, variable decoration rules are never applied and t cannot be a variable, too.

Proposition 11.7 says, that there is a proof with all rule applications at Λ at the end. Since this is the case concerning ϕ and ϕ', we can assume their application to be in this final proof sequence. Let

$$t_m{}^{:\emptyset} =_d t_m{}^{:S_m} \xrightarrow{\Lambda,\phi_m,\sigma_m} t_{m+1}{}^{:S_{m+1}} \xrightarrow{\Lambda,\phi_{m+1},\sigma_{m+1}} \cdots \xrightarrow{\Lambda,\phi_n,\sigma_n} t_{n+1}{}^{:S_{n+1}} =_d t^{:S\cup\{A,B\}}$$

be this sequence. Note that no application of the rules at Λ changes any decoration of strict subterms. Therefore we have $t_i{}^{:\emptyset} \cong_d t_{i+1}{}^{:\emptyset}$ for $i \in [m..n]$, giving us a $\mathcal{T}$-unifier of $p^{:\emptyset}$ and $p'^{:\emptyset}$. Remark that $t_i{}^{:\emptyset}$ must be in $\mathcal{T}$ by the D-closure of $\mathcal{T}$. From Lemma 9.8 now follows that $\mathbf{UNIF_d}$ must succeed, since it is assumed to be $strict\_subterm\_set(\mathcal{T})$-complete and $p^{:\emptyset} \cong_d^? p'^{:\emptyset}$ is a unification problem in $CP(D,D)_{|\mathcal{T}}$, because of $p, p' \notin \mathcal{X}_\Diamond$

Last but not least, there cannot exist a subsort C of A and B in T or T', since decoration rewriting always increases the decorations, i.e. C would therefore also be in S, a clear contradiction to our minimality assumption.

Consequently, we must have a proof of $t' \downarrow_D =_d t^{:S\cup\{A,B\}}$ with a final sequence containing two rule applications on top adding A and B, such that the left hand sides of the rules are unifiable using $\mathbf{UNIF_d}$, i.e. the conditions for the application of **Detect** are fulfilled.

$\Rightarrow$: If **Detect** is applicable, then take $\psi(p^{:\emptyset})^{:S\cup S'\cup\{A,B\}}$ as counter example for sort inheritance. □

Finally, we show how to achieve the $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-fairness for decoration rules with the **Deduce_DD** rule.

Corollary 11.9 *Let $(D, E, R) = (D_k, E_k, R_k)$ be a decorated presentation obtained by a derivation using rules in $\mathcal{D}ed$, s.t. all critical pairs in D_k have been solved. If* **Detect** *cannot be applied, then*

1. *(D, E, R) is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-sort inheriting w.r.t. $\leq_{\mathcal{S}}^{syn}$,*
2. *the derivation is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-fair for decoration rules and*
3. *D is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-confluent.*

Proof: Let us define $\mathcal{T}^n = reach_D(\{t^{:\downarrow\emptyset} \mid |t| \leq n\})$ for $n \geq 0$, which implies $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset}) = \bigcup_{n\geq 0} \mathcal{T}^n$.

The proof is by induction over n. In the case $n = 0$, we have $\mathcal{T}^0$-fairness, since $strict\_subterm\_set(\mathcal{T}^0)$ is empty and $\mathbf{UNIF_d}$ is therefore trivially $strict\_subterm\_set(\mathcal{T}^0)$-complete. Propositions 11.5 and 11.8 now give the $\mathcal{T}^0$-confluence of D and the $\mathcal{T}^0$-sort inheritance respectively.

For $n > 0$ we have $strict\_subterm\_set(\mathcal{T}^n) \subseteq \mathcal{T}^{n-1}$, since $strict\_subterm\_set(reach_D(\mathcal{T}^{n:\downarrow\emptyset})) \subseteq reach_D(strict\_subterm\_set(\mathcal{T}^{n:\downarrow\emptyset})) = reach_D(\mathcal{T}^{n-1:\downarrow\emptyset})$,and therefore the induction hypothesis gives us with the $\mathcal{T}^{n-1}$-sort inheritance w.r.t. $\leq_S^{syn}$ also the $strict\_subterm\_set(\mathcal{T}^n)$-completeness of $\mathbf{UNIF_d}$ (by Proposition 6.10), implying as before $\mathcal{T}^n$-confluence of D resp. $\mathcal{T}^n$-sort inheritance w.r.t. $\leq_S^{syn}$.

Consequently, (D, E, R) is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-sort inheriting w.r.t. $\leq_S^{syn}$ and therefore and it follows once more that $\mathbf{UNIF_d}$ is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-complete, the derivation is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-fair for decoration rules and D is $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-confluent. □

Since we now have a sufficient test for $reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\downarrow\emptyset})$-sort inheritance w.r.t. $\leq_S^{syn}$, the next step is to prove that the test is also sufficient test for $Valid\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sort inheriting. In order to do that, the saturation process must incorporate equalities and rewrite rules.

11.2 Sort Inheritance on Valid Decorated Terms

The general problem in the following will be the test of sort inheritance over all valid terms. Clearly, the validity of a term depends on the provable equalities in the current presentation $\mathcal{P}$, which are decidable when we have a confluent decorated term rewriting system. This can only be achieved with a complete unification algorithm. But testing the completeness for all valid terms is exactly the problem we want to solve.

In order to break this interdependence, we try to achieve confluence with additional restrictions. Several possibilities are considered. The first and easiest one is the restriction of the allowed term declarations. However, we loose a lot of expressiveness in G-algebra when this part of the language is weakened. Another possibility is the design of another proof transformation, that normalizes proofs but keeps the typability property of terms. This leads to sophisticated completion strategies. Therefore, the expressiveness of term declarations can be expensive in practice and a compromise between complexity of completion and unrestricted term declarations is necessary.

The underlying plan for achieving confluence and testing sort inheritance will be the construction of a proof transformation that preserves the *typability* of all decorated terms in a proof, i.e. if for all decorated terms $t^{:S}$ in the proof to be transformed, $t^{:\downarrow\emptyset} \overset{*}{\longleftrightarrow}_D t^{:S}$ holds, then this is also true for all terms in the transformed proof. A closer look at the proof of Theorem 8.1 reveals the following Lemma.

Lemma 11.10 *Let* $\Psi : t_0{}^{:\downarrow\emptyset} \longleftrightarrow_{D_0 \cup E_0} t_1{}^{:S_1} \longleftrightarrow_{D_0 \cup E_0} \ldots t_n{}^{:S_n}$ *be the result of the proof transformation described in the proof of theorem 8.1. Then*

$$\forall i \in [1..n] \; : \; t_i{}^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{D_0} t_i{}^{:S_i}.$$

Proof: This corresponds with property H_2 shown in the proof of Theorem 8.1. □

Using this observation, we can therefore test sort inheritance on all terms in the current proofs (using the results of subsection 11.1) leading to the completeness of $\mathbf{UNIF_d}$ for critical pairs calculation.

We present and discuss three alternatives in the following, after proving typing proof conservation of our matching and unification algorithms. The first proposition is the standard decorated rewriting and completion using flat and linear term declarations, also called function declarations. Dropping the linearity condition forces us to reduce terms in parallel, s.t. all identical subterms at the same depth are reduced simultaneously. This *layer rewriting* extends the set of critical pairs considerably, but does not lead to further restrictions on the completion strategy. Last but not least, we investigate

term declarations without any structural conditions. Then the proof of typability of terms in proofs is possible thanks to maximally subterm sharing rewriting, where all identical redexes have to be reduced at the same time. The principal drawback of this generality is a strong strategy for the completion procedure. However, the needed critical pairs are simpler than for layer rewriting.

Finally, the three approaches are briefly compared and conservative extensions of the initial presentation $\mathcal{P}$ are given for the case when **Detect** applies.

11.3 Subterm Conservation and Typing Proofs

We show here a preliminary property of typability of terms in a match or a unifier of typable terms. Remember that σ_{nf} denotes the tree-solved form corresponding to a solution σ in dag-solved form. The notation $C[\sigma(\Upsilon)]$ is used to denote the proof obtained from the proof Υ by instantiating it with the substitution σ and putting the result into a term context C. Concatenation of proofs is denoted by $+$. For any bottom-up proof Υ, $\Upsilon_{|\omega}$ is obtained by considering subterms at position ω for all terms in Υ and restricting to rule applications on these subterms. These algebraic notations are consistent with [Bac91].

Lemma 11.11 *Let $\sigma = \{x_i{}^{:S_i} \mapsto u_i{}^{:U_i}\}_{i\in I}$ be a decorated substitution in dag-solved form, s.t. $\forall i \in I$: $u_i{}^{:\downarrow\emptyset} \stackrel{*}{\rightarrowtail}_D u_i{}^{:U_i}$ $(H(u_i{}^{:U_i}))$ (resp. $u_i{}^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D u_i{}^{:U_i}$). Then, for all $z \in \mathcal{D}om(\sigma)$:*

$$\sigma_{nf}(z)^{:\downarrow\emptyset} \stackrel{*}{\rightarrowtail}_D \sigma_{nf}(z)) \qquad (H(z))$$

$$(\text{resp. } \sigma_{nf}(z)^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D \sigma_{nf}(z)) \qquad (H'(z))).$$

Proof: The proof is by induction over the occur check ordering of the variables in $\mathcal{D}om(\sigma)$. Let therefore $oc(z)$ be the minimal $k \geq 1$, s.t. $\sigma^k(z) =_d \sigma^{k+1}(z)$. Note that $oc(z) = 0$ is obvious, since this implies that $z =_d \sigma(z) =_d \sigma_{nf}(z)$.

Else let $oc(z) = k > 1$, $\sigma(z) = u_n{}^{:U_n}$, $\mathcal{V}ar(\sigma(z)) = V = \{z_j \mid j \in J\}$ and O_j denote the set of occurrences of z_j in $\sigma(z)$ for all $j \in J$. We can suppose $H(z_j)$ to be given for all $j \in J$ by the induction hypothesis. Let us denote by Υ^j the corresponding typing proof of $\sigma_{nf}(z_j)$. Then

$$\sigma_{nf}(z)^{:\downarrow\emptyset}[\Upsilon^j]_{O_j, j\in J} \; : \; \sigma_{nf}(z)^{:\downarrow\emptyset} \stackrel{*}{\rightarrowtail}_D \sigma_{nf}(z)^{:\downarrow\emptyset}[\sigma_{nf}(z_j)]_{O_j, j\in J}.$$

Now, using $H(u_n{}^{:U_n})$ and substitutivity of decoration rewriting gives us

$$\Upsilon^n \; : \; (\sigma_{nf})_{|V}(\sigma_{nf}(z)^{:\downarrow\emptyset}) =_d \sigma_{nf}(z)^{:\downarrow\emptyset}[\sigma_{nf}(z_j)]_{O_j, j\in J} \stackrel{*}{\rightarrowtail}_D \sigma_{nf}(z).$$

Concatenating $\sigma_{nf}(z)^{:\downarrow\emptyset}[\Upsilon^j]_{O_j, j\in J}$ with Υ^n now results in $H(z)$. The case $H'(z)$ is similar. □

Proposition 11.12 *Let $\mathcal{M}$ be a matching problem, s.t. for all $t \in terms(\mathcal{M})$, $t^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D t$ and $\sigma = \{x_i{}^{:S_i} \mapsto u_i{}^{:U_i}\}_{i\in I}$ the $\mathcal{D}$-matcher calculated by* $\mathbf{MATCH_d}$ *in dag-solved form. Then:*

$$\forall i \in I \; : \; \sigma_{nf}(x_i{}^{:S_i})^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D \sigma_{nf}(x_i{}^{:S_i})$$

Proof: Clearly, the strict subterm conservation implies that the $u_i{}^{:U_i}$ are subterms of some $t \in terms(\mathcal{M})$. Now 11.11 guarantees that σ_{nf} has the desired property. □

Proposition 11.13 *Let $\mathcal{U}$ be a unification problem, s.t. for all $t \in terms(\mathcal{U})$, $t^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D t$ (resp. $t^{:\downarrow\emptyset} \stackrel{*}{\rightarrowtail}_D t$) and $\sigma = \{x_i{}^{:S_i} \mapsto u_i{}^{:U_i}\}_{i\in I}$ the $\mathcal{D}$-unifier calculated by* $\mathbf{UNIF_d}$ *in dag-solved form. Then for all $t \in terms(\mathcal{U})$:*

$$\sigma_{nf}(t)^{:\downarrow\emptyset} \stackrel{*}{\leftarrowdiamond\rightarrow}_D \sigma_{nf}(t)) \qquad (H(t))$$

$$(\text{resp. } \sigma_{nf}(t)^{:\downarrow\emptyset} \stackrel{*}{\rightarrowtail}_D \sigma_{nf}(t))) \qquad (H'(t)).$$

Proof: We know that decorated unification is subterm conservative. Let τ be the variable renaming for some $u_i{}^{:U_i}$ with $i \in I$, i.e. for all $x^{:S_x} \in \mathcal{D}om(\tau)$, we have $\tau(x^{:S_x}) =_d z^{:S}$ for some $z \in \mathcal{X}_\Diamond$ and $sort(z) \approx S$. Clearly, $z^{:\downarrow\emptyset} =_d z^{:\{C\}} \rightarrowtail_D^* z^{:S}$, since $z^{:\{C\}} \cong_d z^{:S}$.

Hence,

$$\forall i \in I \ : \ (u_i{}^{:U_i})^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* u_i{}^{:U_i},$$

by substitutivity of decoration rewriting, since $u_i{}^{:U_i} =_d \tau(t_{|\omega})$ for some $t \in terms(\mathcal{U})$, which makes 11.11 applicable, giving us $H(t)$ for all $t \in terms(\mathcal{U})$ once more by substitutivity. The proof of $H'(t)$ is similar. □

11.4 Flat and Linear Term Declarations

Restricting to flat and linear term declarations, it is possible to prove that typability of terms is preserved by standard rewriting, as defined in section 7.

Lemma 11.14 *Let D be a set of decoration rewrite rules, that contains flat and linear decoration rules only and R be a set of decorated rewrite rules, s.t. all terms $p^{:S}$ in R satisfy $p^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* p^{:S}$. Let furthermore $t^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* t^{:T}$ and $t^{:T} \rightarrowtail_R t'^{:T'}$.*

Then $t'^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* t'^{:T'}$.

Proof: We can construct the new typing proof by the strict subterm conservation of decorated matching(see proposition 11.12), the substitutivity of the typing proof for the right hand side of the rule used for simplification and at any occurrence incomparable or above the reduction, we can use the old proof unchanged. Formally, if $\Psi : t^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* t^{:T}$ is a typing proof for the term to be reduced, $\Psi' : r^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* r^{:S_r}$ and $t^{:T}[\sigma(r^{:S_r})]_\omega$ is the reduced term, then:

$$t^{:\downarrow\emptyset}[\Psi_{|\omega.\mathcal{VO}cc(l)}]_{\omega.\mathcal{VO}cc(r)} + t^{:\downarrow\emptyset}[\sigma(\Psi')]_\omega : (t^{:T}[\sigma(r^{:S_r})]_\omega)^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_D^* (t^{:T})^{:\downarrow\emptyset}[\sigma(r^{:S_r})]_\omega$$

where we can append all steps of Ψ that apply strictly above ω. □

Actually, the property of being flat and linear is only required for D_∞.

Lemma 11.15 *Let $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using $\mathcal{OSC}$. Then for all $k \geq 0$, all terms that are typable in D_k are also typable in D_∞, if D_∞ only contains flat, linear decoration rewrite rules.*

Proof: Suppose that $\Psi_k : t^{:\downarrow\emptyset} \leftarrow\!\diamond\!\rightarrow_{D_k}^* t^{:S}$ is the typing proof in D_k. Then we can first prove for each decoration rewrite rule ($\phi : l^{:s} \rightarrow l^{:s \cup S_l}$ if $S_l \not\subseteq s$) used in Ψ_k, that it is either in D_∞ or subsumed by some $\phi' \in D_\infty$ in the sense of **Subsume_deco**.

The proof is an induction on the size of $l^{:S_l}$ w.r.t. the strict part $<_d \cup \lesssim_d$ of $\leq_d \cup \lesssim_d$. Remark that this ordering is well-founded, since $<_d$ and $\lesssim_d$ commute [BD86], i.e. $s <_d t \lesssim_d \sigma(t)$ implies $s \lesssim_d \sigma(s) <_d \sigma(t)$. Therefore, any infinite sequence of $<_d \cup \lesssim_d$ can be separated into a starting sequence using $\lesssim_d$ only followed by a pure $<_d$ sequence, in contradiction to their well-foundedness.

If $l^{:S_l}$ is irreducible by $\bigcup_{n \geq k} D_n \cup R_n$, then it must either be subsumed by some $(\phi'' : l''^{:s} \rightarrow l''^{:s \cup S_{l''}}$ if $S_{l''} \not\subseteq s) \in D_m$, $m \geq k$, or it is still in D_∞. In the first case we can apply the induction hypothesis on ϕ'' and get the replacement of $\phi' \in D_\infty$ of ϕ'', that must also subsume ϕ, due to transitivity of subsumption. Remark that if $S_l \subsetneq S_{l''}$, then $l''^{:S_{l''}} <_d l^{:S_l}$, since $l''^{:S_{l''}} <_d l''^{:S_l}$ and $l''^{:S_l} \lesssim_d l^{:S_l}$.

Otherwise, if $S_l \approx S_{l''}$, and $l''^{:S_{l''}} \leq_d l''^{:S_l} \lesssim_d l^{:S_l}$.

If $l^{:S_l}$ is reduced at step n with R_n, then the top symbol remains the same and therefore using the induction hypothesis on the reduced term gives a ϕ' that is flat and linear. Hence, ϕ' is also subsuming ϕ. The same argument can be applied in the case of $l^{:S_l}$ reduced with D_n. Note that in both cases $l''^{:S_{l''}} <_d l^{:S_l}$.

Now, let Ψ_∞ be the proof $t^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D_\infty} t^{:S}$, where every step $\overset{\phi}{\rightarrowtail}_{D_k}$ in Ψ_k is replaced by $\overset{\phi'}{\rightarrowtail}_{D_\infty}$ o $\overset{0.1\phi'}{\leftarrowtail}_{D_\infty}$, i.e. any rule is replaced by the corresponding one in D_∞. □

We get the following results, where CT^k_{DER} denotes the set of all decorated terms in $D_k \cup E_k \cup R_k$ of a decorated presentation (D_k, E_k, R_k) during completion, where sort set variables s in decoration rules are replaced by $\emptyset$.

Lemma 11.16 *Let $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using OSC. Then for all $k \geq 0$, all terms in CT^k_{DER} are typable in D_∞, if D_∞ only contains flat, linear decoration rewrite rules.*

Proof: The proof is an induction over the number of completion steps k. If $k = 0$, then all $t^{:S} \in CT^0_{DER}$ are equal to $t^{:\downarrow\emptyset}$ resp. $(t^{:\downarrow\emptyset})^{:S}$ for terms in decoration rewrite rules and therefore trivially typable.

For $k > 0$, we can construct a typing proof of a newly added term in any case. Remark that completion rules adding new terms to CT^k_{DER} do not change or extend the set of decoration rules. If the new term is obtained by decoration rewriting from an old one, the claim is trivial.

If it is obtained by decorated rewriting, we can use Lemma 11.14 in order to construct the new typing proof. If the new term is obtained by orientation of an equation, then it can be typed with the proof of the old term in the equation plus, eventually, the new decoration rule applied on top in the end.

Finally, the new term can be the result of a critical pair computation. This can be seen as the unification of two terms $t^{:S}, t'^{:S'}$ in CT^{k-1}_{DER}. The resulting unifier ψ applied to one of the terms in CT^{k-1}_{ER}, gives again a typable term, because of proposition 11.13 and stability of decoration rewriting under substitutivity used for the typing proof of the initial term. The second step is the reduction of $\psi(t^{:S})$ with the lower rule, where Lemma 11.14 can be used.

Finally, there are completion rules manipulating decoration rules. In this case, Lemma 11.14 gives us a corresponding typing proof in D_∞ that does not change. □

Theorem 11.17 *Let $\mathcal{P}_\infty \neq (\bot, \bot, \bot)$ be the presentation obtained from $(D_\mathcal{P}, E_\mathcal{P}, \emptyset)$ using OSC, s.t. all terms in D_∞ are flat and linear. Let furthermore $E_\infty = \emptyset$ and all critical pairs of $D_\infty \cup R_\infty$ be in $D_* \cup R_*$. Then the initial presentation $\mathcal{P}_0$ is sort inheriting on $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ and $D_\infty \cup R_\infty$ is Church-Rosser, type and existentially complete.*

Proof: The fact that all decoration rewrite rules in D_∞ are flat and linear implies that D_∞ is confluent and therefore $\mathcal{P}_\infty \neq (\bot, \bot, \bot)$ gives us sort inheritance on all typable terms. Furthermore, any term generated by the proof reduction $\Longrightarrow$ is typable in D_∞, because of Lemmas 11.16, 11.14 and 11.15. Remark that any new term in a proof generated by $\Longrightarrow$ at completion step k is reachable from an old one by $\overset{*}{\rightarrowtail}_{D_{k+1} \cup R_{k+1}}$.

Hence, sort inheritance on the set $reach_{D_\infty}(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond))$ gives us completeness of $\mathbf{UNIF_d}$ needed for peak reduction. Furthermore, the condition that all critical pairs of $D_\infty \cup R_\infty$ be in $D_* \cup R_*$ and $E_\infty = \emptyset$ prove the fairness of the derivation (see proposition 10.8) and therefore $D_\infty \cup R_\infty$ has to be Church-Rosser.

Consequently, all terms in normal forms of proofs constructed by theorem 8.1 are typable and do not contain any peaks. Therefore, all minimal sorts to which a term belongs can be found in the top decoration of its normal form: Suppose this is not the case for $\overline{t}^{:\downarrow\emptyset}$. Then let $t^{:S}$ be its $D_\infty \cup R_\infty$-normal form, that must be typable, $\Psi : \overline{t}^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_{D_\infty \cup R_\infty} t'^{:S'}$ a proof implying $t : A$ with $\nexists B \in S : B \leq_S^{syn} A$, i.e. w.l.o.g. $A \in S'$. Consequently, we have a normal form Ψ_{nf} of Ψ, s.t.

$$\Psi_{nf} : \overline{t}^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{D_\infty \cup R_\infty} t''^{:S''} \overset{*}{\leftarrowtail}_{D_\infty \cup R_\infty} t'^{:S'},$$

where $t''^{:S''}$ is irreducible. Now, $t^{:S} \overset{*}{\leftarrowtail}_{D_\infty \cup R_\infty} \overline{t}^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_{D_\infty \cup R_\infty} t''^{:S''}$ is a proof with typable terms only, that is reduced by $\Longrightarrow$ into a rewrite proof, i.e. $t^{:S} \cong_d t''^{:S''}$ since both terms are supposed to be irreducible. Therefore, $S \approx S''$, in contradiction to $A \in S'$ and $\nexists B \in S : B \leq_S^{syn} A$.

The type and existential completeness follow now from the fact that decoration and decorated rewriting never decrease the top decoration. The typability of the normal form allows us to extend the result of the sort inheritance test to the whole set of valid terms $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, i.e. $\mathcal{P}_0$ is sort inheriting on $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, because of proposition 10.3 and the fact that we have equal sorts for equal terms. □

11.5 Flat Term Declarations

In order to relax the restriction of linearity on term declarations, the idea is to replace rewriting by layer rewriting. At the price of trickier proofs and more critical pair computations, we can prove an extension (Theorem 11.29) of Theorem 11.17.

11.5.1 Definition and Simple Properties

Definition 11.18 *Let $t^{:S}, t'^{:S'} \in \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $\phi \in R$ and k be a natural number.*

Then $t^{:S} \rightleftarrows_R^{\phi,\sigma,k} t'^{:S'}$, if $t^{:S} \rightarrowtail_R^{\phi,\sigma,O} t'^{:S'}$, s.t. $\forall \omega \in O$, $|\omega| = k$ and O is maximal. Given ϕ, σ and k, $O_{(t^{:S},\phi,\sigma,k)}$ stands for such an O.

Instead of $t[u]_{O_{(t[u],\phi,\sigma,k)}}$ we simply write $t[u]_{O_{(\phi,\sigma,k)}}$.Layer rewriting has the advantage that it preserves typability of terms by flat (not necessarily linear) term declarations, which correspond with the so-called semi-linear membership theories. It has the disadvantage of the maximality condition, which destroys stability by context and substitution.

Lemma 11.19 *Let D be a set of decorated rewrite rules with flat terms only and R a set of decoration rewrite rules. Let furthermore $t^{:S} \rightleftarrows_R^{\phi,\sigma,k} t'^{:S'}$ and $(\phi : l^{:S_l} \rightarrow r^{:S_r}) \in R$, s.t. there are typing proofs $\Psi : (r^{:S_r})^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_D r^{:S_r}$ and $\Psi' : (t^{:S})^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_D t^{:S}$.*

Then there is a proof $\Upsilon : (t'^{:S'})^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_D t'^{:S'}$.

Proof: First, notice that we can assume Ψ, Ψ' to be bottom-up w.l.o.g., since there are no decoration critical pairs. Clearly, $t'^{:S'} \cong_d t^{:S}[\sigma(r^{:S_r})]_{O_{(\phi,\sigma,k)}}$. Furthermore, Lemma 11.12 guarantees us the typability of the terms in $\mathcal{I}m(\sigma)$ and therefore we get by substitutivity and context stability of decoration rewriting:

$$(t^{:s})^{:\downarrow\emptyset}[\sigma(r^{:S_r})^{:\downarrow\emptyset}[\Psi'_{|_{O_{(\phi,\sigma,k)}.\mathcal{VOcc}(l^{:S_l})}}]_{\mathcal{VOcc}(r^{:S_r})+\sigma(\Psi)}]_{O_{(\phi,\sigma,k)}} : (t'^{:S'})^{:\downarrow\emptyset} \overset{*}{\leftrightarrow}_D (t^{:S})^{:\downarrow\emptyset}[\sigma(r^{:S_r})]_{O_{(\phi,\sigma,k)}}$$

Now we can append also all decoration rule applications at occurrences incomparable to $O_{(\phi,\sigma,k)}$ in Ψ', since their substitutions do not change. For all rule applications above some $\omega \in O_{(\phi,\sigma,k)}$, we only have to adapt the substitutions, since the rules are flat and $O_{(\phi,\sigma,k)}$ is maximal. Consequently, Υ exists.

□

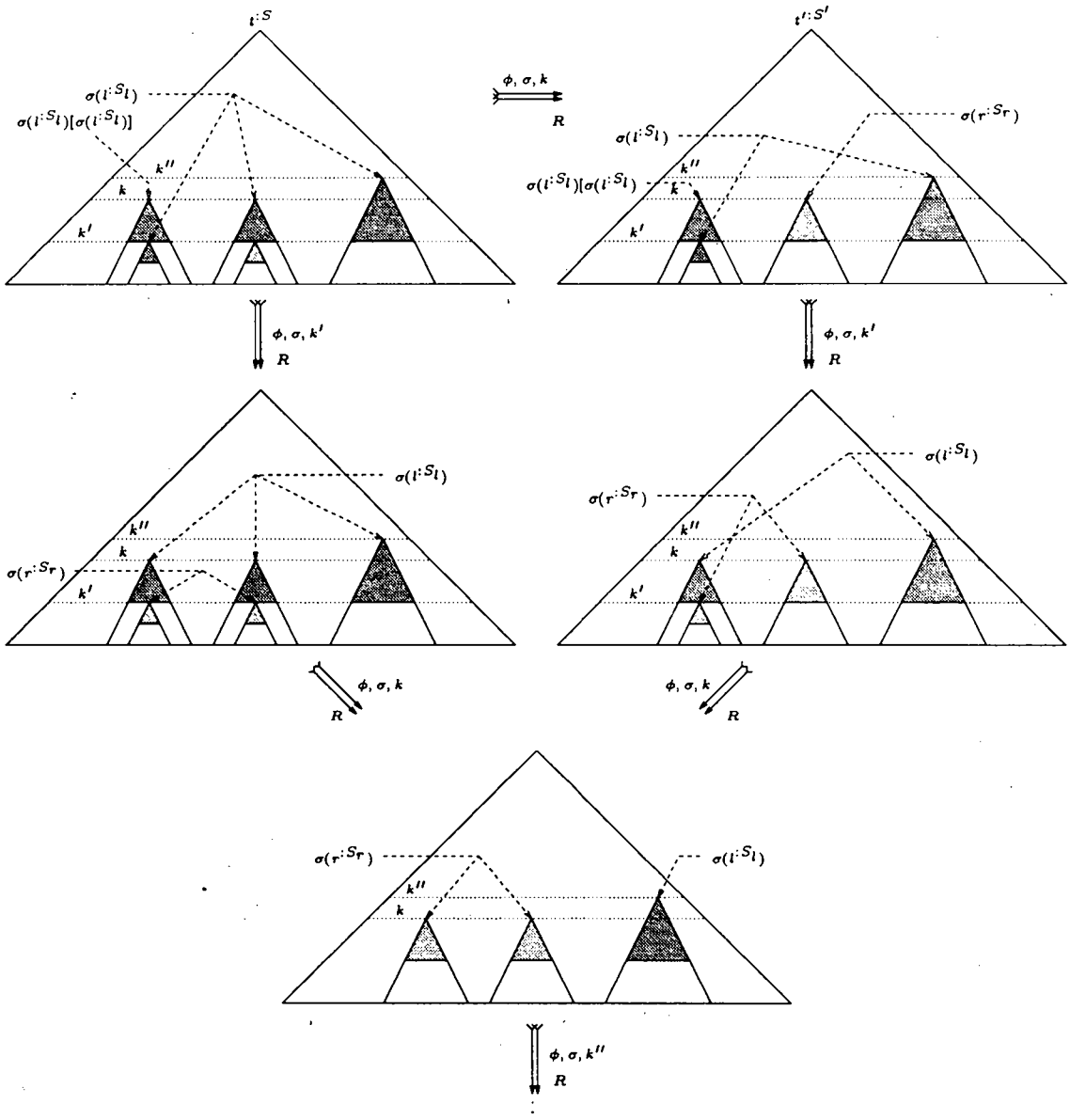


Figure 11: Bottom-up Layer Rewriting

Another property is needed in order to get unique reduced forms.

Lemma 11.20 *Let* $t^{:S} \rightrightarrows_R^{\phi,\sigma,k} t'^{:S'}$. *Then*

$$t^{:S} \overset{*}{\rightrightarrows}_R^{\phi,\sigma} t''^{:S''} \overset{*}{\leftrightarrows}_R^{\phi,\sigma} t'^{:S'},$$

where the two rewrite sequences are bottom-up for all layers strictly deeper than k'' *for some* $k'' < k$.

Proof: This follows immediately from the fact that a layer rewrite step at layer k only introduces redexes for the same pair (ϕ, σ) that are situated on a layer l with $l < k$ in the reduced term. Consequently, the subterms at the mounting layers k' are identical in the two rewrite sequences, as long as $k' < k$. At k, the left rewrite sequence reduces the redexes that were also reduced in the step $t^{:S} \rightrightarrows_R^{\phi,\sigma,k} t'^{:S'}$ plus those generated by the steps in deeper layers l, i.e. $l > k$, whereas the right rewrite sequence reduces only the latter. Remark that there are no additional redexes for (ϕ, σ) resulting from $t^{:S} \rightrightarrows_R^{\phi,\sigma,k} t'^{:S'}$, since these have to be on a layer l with $l < k$. Therefore, the left and the right rewrite sequence are identical for all layers $k' < k$ up to k''. This is illustrated in figure 11, where we assume ϕ to be $l^{:S_l} \to r^{:S_r}$. □

In the following definition, we need the notion of ascending narrowing (in R) beginning with some occurrence ω. This denotes the 'classical' narrowing steps along the occurrences of the narrowed term strictly above ω up to $\wedge$, s.t. every step is done strictly above all preceding ones. This extends canonically to occurrence sets O.

These narrowing steps will be necessary in order to adjust substitution images in the Critical Pair Lemma 11.24, since decorated rewriting rules are not necessarily flat or linear. Therefore, reducing all redexes for some $\phi \in R$ in a term t in a layer below a redex for some $\phi' \in R$ might change the image of a variable x in the domain of σ, the matcher needed to apply ϕ'. Hence, there may be no σ' that could be used to apply ϕ' on the changed term t'. So we need to equalize the subterms that were equal in t but are no more equal in t', in order to find such a σ'. But these may be situated on different layers. In order to get a unique reduced form, we use a bottom-up strategy for the equalization together with Lemma 11.20.

In some cases, it will be necessary to mount up to the left hand side l of ϕ', since some non-variable parts of l may be situated on deeper layers as the variable x. Consequently, critical pairs have to take care of such bottom-up reductions, s.t. mounting layer reductions correspond with mounting narrowing steps.

Furthermore, we need to combine unifiers $\{\sigma_i\}_{i \in I}$. This is done by retransforming each unifier $\sigma_i = \{x_{ij}{}^{:S_{ij}} \to t_{ij}{}^{:S'_{ij}}\}_{j \in J_i}$ to be combined into a unification problem $(\sigma_i)_= = \bigwedge_{j \in J_i} x_{ij}{}^{:S_{ij}} \cong_d^? t_{ij}{}^{:S'_{ij}}$ (in solved form). Then the result of the combination is the $\mathcal{D}$-solution ψ for the problem $\bigwedge_{i \in I}(\sigma_i)_=$.

Definition 11.21 Layer critical pairs (obtained by layered superposition *into* decorated rules)
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $g^{:S_g} \to d^{:S_d}$ *and* $l^{:S_l} \to r^{:S_r}$ *be rewrite rules in* R *with disjoint sets of variables.*

The two rules overlap if there exists a position ω *in the set of non-variable positions of* $g^{:S_g}$, *such that the decorated terms* $g^{:S_g}{}_{|\omega}$ *and* $l^{:S_l}$ *have the* $\mathcal{T}$*-complete,* $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$*-sound, non-empty set* Ψ *of strict decorated unifiers.*

Let $O = \{\omega_i\}_{i \in [1..n]}$ *be the set of such positions and* Ψ_i *the corresponding unifiers. Let furthermore* $O' = \{\omega_j\}_{j \in [1..m]}$ *be the set of all non-variable overlap positions of* $l^{:S_l}$ *into* $d^{:S_d}$ *and* Ψ'_j *the corresponding unifiers.*

Then for any $i \in [1..n]$, *any combination* ψ *of some unifiers* $\{\sigma \mid \exists i \in [1..n] : \sigma \in \Psi_i$ or $existsj \in [1..m] : \sigma \in \Psi'_j\}$ *corresponding with the overlap positions* $\overline{O} \subseteq O$ *and* $\overline{O'} \subseteq O'$, *s.t.* $\overline{O} \cap O \neq \emptyset$ *and all occurrences in* $\overline{O} \cup \overline{O'}$ *are incomparable, the overlap produces the* $\mathcal{T}$*-layer critical pair (*$\mathcal{T}$*-LCP)* $(p^{:S_1} = q^{:S_2})$ *where* $q^{:S_2} =_d \psi(d^{:S_d}[r^{:S_r}]_{\overline{O'}})$ *and* $p^{:S_1} =_d \psi(g^{:S_g}[r^{:S_r}]_{\overline{O}})$.

A narrowed $\mathcal{T}$*-LCP is a decorated equation* $p'^{:S'_1} = q'^{:S'_2}$, *s.t. there is a* $\mathcal{T}$*-LCP* $(p^{:S_1} = q^{:S_2})$ *and* $p'^{:S'_1}$ *resp.* $q'^{:S'_2}$ *can be obtained from* $p^{:S_1}$ *resp.* $q^{:S_2}$ *via ascending narrowing.*

Let $LCP(R,R)_{|_\tau}$ denote the set of such layer critical pairs. Clearly, these sets of critical pairs are finite, since the set of positions of a term is finite.

Example 11.22 *Let* $(\phi : g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} \to g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$ *and*
$(\phi' : f(z^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle} \to z^{:\langle A\rangle})$ *be decorated rewrite rules.*

Clearly, ϕ' *overlaps into* ϕ *at occurrences* 1, 2 *and* 2.1 *with unifiers* $\sigma_1 = \{z^{:\langle A\rangle} \mapsto x^{:\langle A\rangle}\}, \sigma_2 = \{z^{:\langle A\rangle} \mapsto f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle} \mapsto a^{:\langle A\rangle}\}$, *respectively* $\sigma_{2.1} = \{z^{:\langle A\rangle} \mapsto y^{:\langle A\rangle}\}$. *Then the narrowed LCPs are:*

- *overlap at* 1: $g(x^{:\langle A\rangle}, f(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$,
  Narrowing at 2 *with* ϕ' *gives :*
  $g(x^{:\langle A\rangle}, f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$,
  Narrowing at 2.1 *gives :*
  $g(x^{:\langle A\rangle}, f(y^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$,
  Now, we can narrow once more at 2*:*
  $g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$,
- *overlap at* 2: $g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$
- *overlap at* 2.1*:*
  $g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(z^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, z^{:\langle A\rangle})^{:\langle B\rangle})$,
  Now, we can narrow at 2*, giving:*
  $g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, z^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, z^{:\langle A\rangle})^{:\langle B\rangle})$,
- *Combining* 1 *and* 2*:*
  $g(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$
  Therefore, we can narrow at 1 *and* 2 *at the same time, since the same substitution can be used. Furthermore, the same redex can be found at* 1 *at the right hand side. We get:*
  $g(y^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\{B\}} = g(y^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$
- *Combining* 1 *and* 2.1*:*
  $g(x^{:\langle A\rangle}, f(x^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, x^{:\langle A\rangle})^{:\langle B\rangle})$,
  Narrowed at 2*:* $g(x^{:\langle A\rangle}, x^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, x^{:\langle A\rangle})^{:\langle B\rangle})$,

The redexes 2 *and* 2.1 *cannot be combined, since they are comparable.*

11.5.2 Peak Reduction

Lemma 11.23 *Let* $t'^{:S'} \rightrightarrows_R^{\phi',\sigma',k'} t^{:S} \rightrightarrows_R^{\phi'',\sigma'',k''} t''^{:S''}$, *s.t.* $O_{(t^{:S},\phi',\sigma',k')} \asymp O_{(t^{:S},\phi'',\sigma'',k'')}$.
Then $t'^{:S'} \overset{*}{\rightrightarrows}_R \circ \overset{*}{\rightrightarrows}_R t''^{:S''}$.

Proof: Let $\phi' : l^{:S_l} \to r^{:S_r}$, $\phi'' : g^{:S_g} \to d^{:S_d}$ and $t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(\phi',\sigma',k')}}[\sigma''(g^{:S_g})]_{O_{(\phi'',\sigma'',k'')}}$.
W.l.o.g. we can assume $k' \leq k''$.

There may be redexes for ϕ', σ' that are created by reductions with ϕ'', σ''. Therefore, we have to assume $\sigma'(l^{:S_l}) \cong_d \sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}$ with $|\omega''| = k''$ for all $\omega'' \in O''$, O'' is maximal and $t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_{O''}]_{O'}[\sigma''(g^{:S_g})]_{\overline{O''}}$,
s.t. $O'' \uplus \overline{O''} = O_{(t^{:S},\phi'',\sigma'',k'')}$.
Consequently, $t'^{:S'} \cong_d t^{:S}[\sigma'(r^{:S_r})]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_{O''}]_{O'}[\sigma''(g^{:S_g})]_{\overline{O''}}$ and

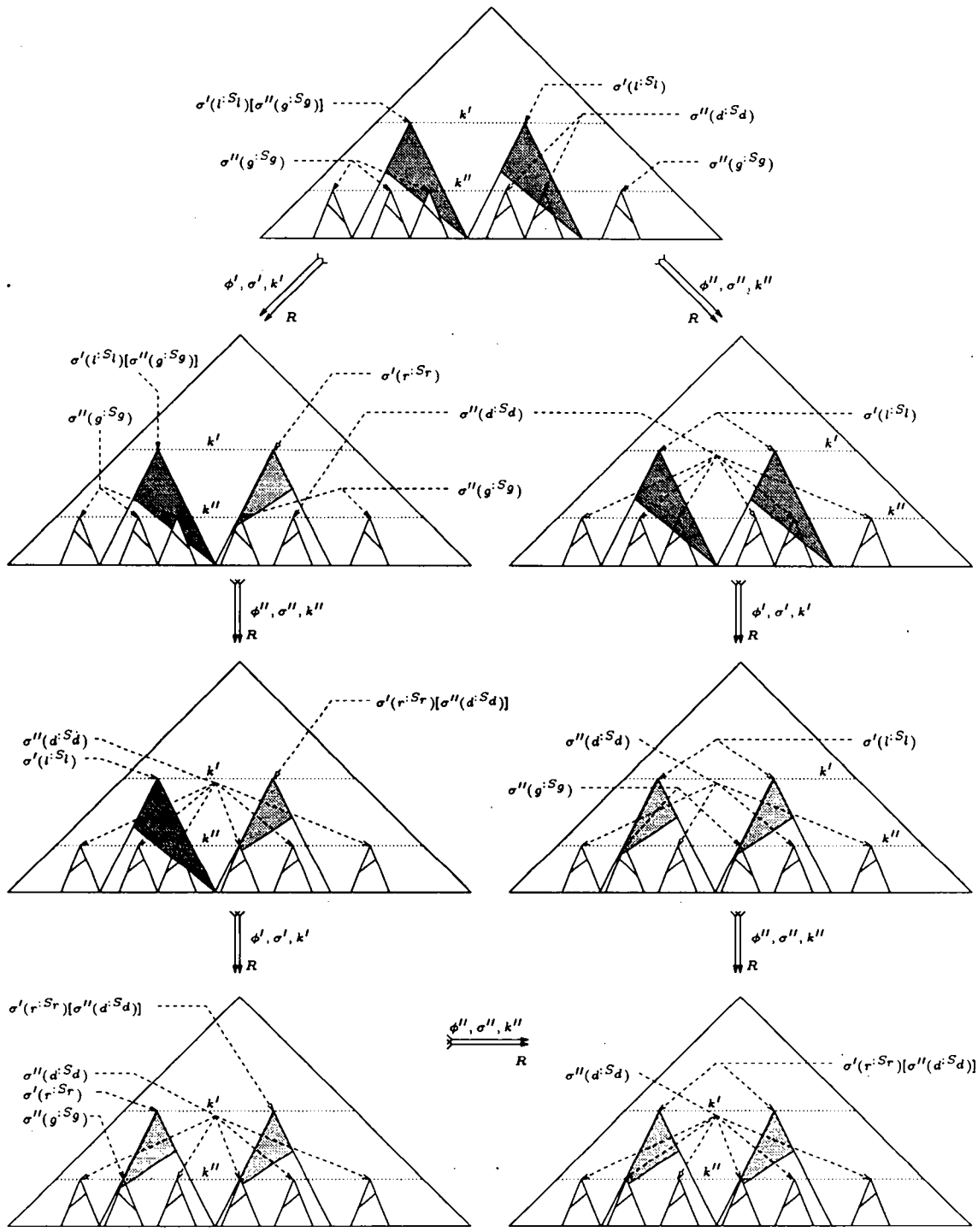


Figure 12: No Overlap Case

$t''^{:S''} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}}$.

Furthermore, the replacement constructed by ϕ' may generate a redex for ϕ'', σ'', i.e.:

$$\begin{array}{rl} & t^{:S}[\sigma'(r^{:S_r})]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_{O''}]_{O'}[\sigma''(g^{:S_g})]_{\overline{O''}} \\ \cong_d & t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_{O''}]_{O'}[\sigma''(g^{:S_g})]_{\overline{O''}} \end{array}$$

Now, we are sure that $O \uplus O'' \uplus \overline{O''} = O_{(t'^{:S'},\phi'',\sigma'',k'')}$ and $O_{(t^{:S},\phi',\sigma',k')} \uplus O' = O_{(t''^{:S''},\phi',\sigma',k')}$. The following peak reduction is illustrated in figure 12:

$$\begin{array}{ll} & t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_{O''}]_{O'}[\sigma''(g^{:S_g})]_{\overline{O''}} \\ \rightrightarrows_R^{\phi'',\sigma'',k''} & t^{:S}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}} \\ =_d & t^{:S}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}} \\ \overset{0,1}{\rightrightarrows}_R^{\phi',\sigma',k'} & t^{:S}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}} \\ \overset{0,1}{\rightrightarrows}_R^{\phi'',\sigma'',k''} & t^{:S}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}} \\ \overset{0,1}{\rightrightarrows}_R^{\phi'',\sigma'',k''} & t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}} \\ \rightrightarrows_R^{\phi',\sigma',k'} & t^{:S}[\sigma'(l^{:S_l})]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_{O'}[\sigma''(d^{:S_d})]_{\overline{O''}}. \end{array}$$

The (optional) 0, 1-steps are due to the cases where either no new redex for the radical ϕ', σ' is generated using ϕ'', σ'' or $k' = k''$, i.e. possibly $\sigma''(d^{:S_d}) \cong_d \sigma'(l^{:S_l})$. □

Lemma 11.24 Layer Critical Pair Lemma
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ be downward complete w.r.t. $\rightrightarrows_R$ and $t^{:S}, t'^{:S'}, t''^{:S''} \in \mathcal{T}$.
Let furthermore $\phi' : l^{:S_l} \to r^{:S_r}$, $\phi'' : g^{:S_g} \to d^{:S_d}$ and $t'^{:S'} \rightrightarrows_R^{\phi',\sigma',k'} t^{:S} \rightrightarrows_R^{\phi'',\sigma'',k''} t''^{:S''}$,
s.t. $k' \leq k''$; $O_{(t^{:S},\phi',\sigma',k')}.\mathcal{NVO}cc(l^{:S_l}) \cap O_{(t^{:S},\phi'',\sigma'',k'')} \neq \emptyset$.
Then $t'^{:S'} \overset{*}{\rightrightarrows}_R \circ \longleftarrow\!\!\circ\!\!\longrightarrow_{LCP(\phi',\phi'')} \circ \overset{*}{\rightrightarrows}_R t''^{:S''}$.

Proof: Let w.l.o.g. $t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(\phi',\sigma',k')}}[\sigma''(g^{:S_g})]_{O_{(\phi'',\sigma'',k'')}}$,
$t'^{:S'} \cong_d t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma''(g^{:S_g})]_{O'}$,
s.t. $O_{(t^{:S},\phi',\sigma',k')}.O \uplus O' = \bigcup_{i \geq k'} O_{(t'^{:S'},\phi'',\sigma'',i)}$ and
$t''^{:S''} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma''(d^{:S_d})]_{O'''}[\sigma''(g^{:S_g})]_{O'\setminus O'''}$,
where $O_{(t^{:S},\phi',\sigma',k')}.O'' \uplus O''' = O_{(t^{:S},\phi'',\sigma'',k'')}$. Hence, O' stands for the redexes of ϕ'' with σ'' 'outside of' $\sigma'(l^{:S_l})$.

Then we can bottom-up reduce up to k':

$$t'^{:S'} \overset{*}{\rightrightarrows}_R^{\phi'',\sigma''} t^{:S}[\sigma'(r^{:S_r})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma''(d^{:S_d})]_{O'}$$

The same is possible for $t''^{:S''}$:

$$\begin{array}{lll} t''^{:S''} & \overset{*}{\rightrightarrows}_R^{\phi'',\sigma''} & t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma''(d^{:S_d})]_{O'''}[\sigma''(d^{:S_d})]_{O'\setminus O'''} \\ & =_d & t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma''(d^{:S_d})]_{O'} \end{array}$$

Since we may have variable overlaps with $l^{:S_l}$, we need to continue with the bottom-up reduction until k'. Remark that in both branches, we have $\sigma''(d^{:S_d})$ at the occurences O', i.e. we can

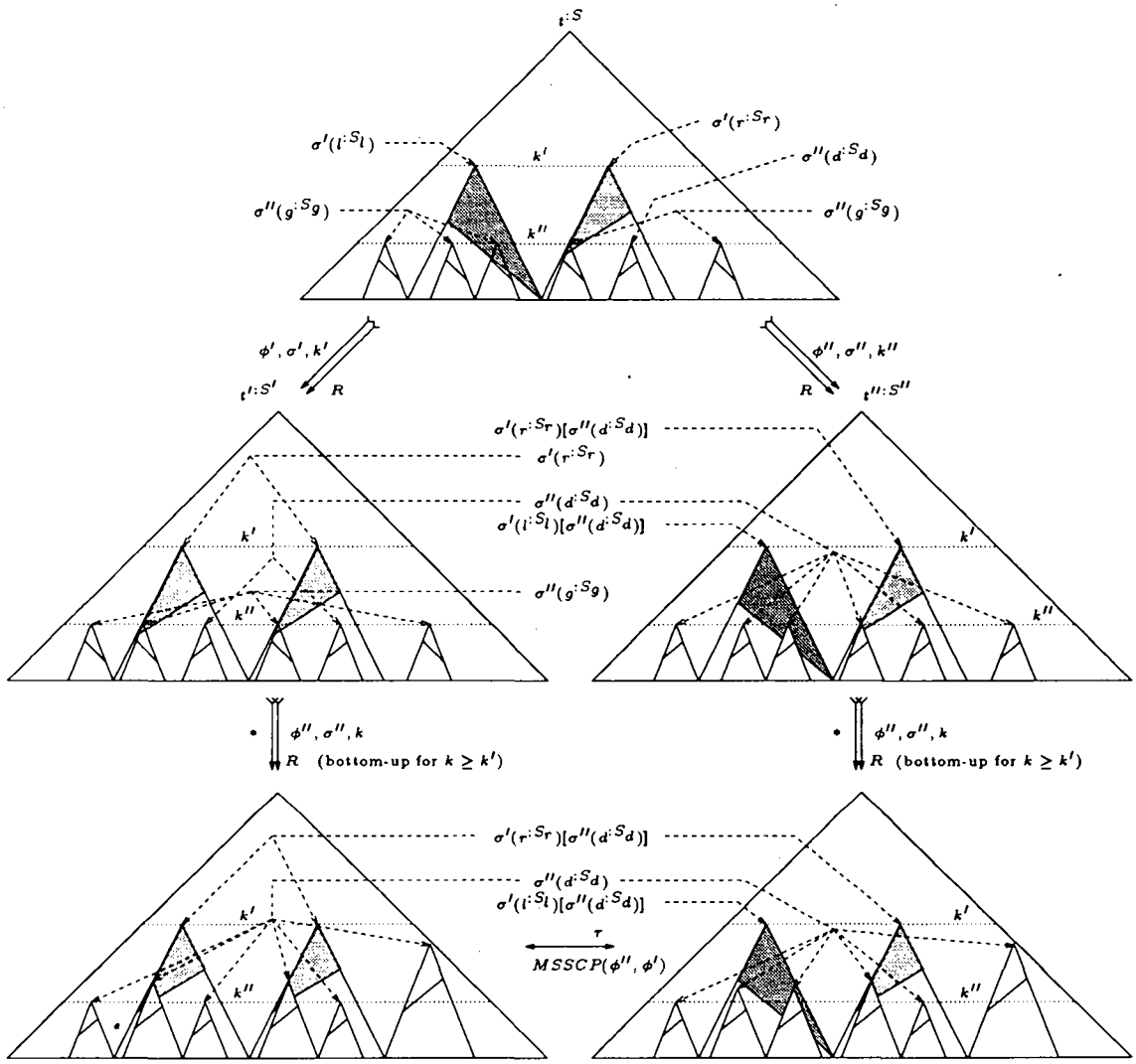


Figure 13: Critical Overlap Case

omit these details in the following. Moreover, the terms $\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}$ and $\sigma'(r^{:S_r})[d^{:S_d}]_O$ correspond with some layer critical pair $\psi(l^{:S_l}[d^{:S_d}]_{O''}) = \psi(r^{:S_r}[d^{:S_d}]_O)$.

Therefore, there exists a narrowed $\mathcal{T}$-LCP $p'^{:S'_1} = q'^{:S'_2}$ that can be applied to the result of the bottom-up reduction until k' of the left and the right reduction sequence.

The narrowed critical pair is actually applicable, since all parts of the unifier ψ used to compute the overlap correspond with strict subterms[2] of the redex $\sigma''(g^{:S_g})$, since we only compute non-variable overlaps. Consequently, we do not need to overlap recursively into the substitution part of $\psi(l^{:S_l})$, which might lead to infinite sets of critical pairs.

□

Only now we are able to prove the variable overlap case, since we may need layer critical pairs in order to solve them.

Lemma 11.25 *Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ be downward complete w.r.t. $\rightrightarrows_R$ and $t^{:S}, t'^{:S'}, t''^{:S''} \in \mathcal{T}$. Let furthermore $\phi' : l^{:S_l} \to r^{:S_r}$, $\phi'' : g^{:S_g} \to d^{:S_d}$ and $t'^{:S'} \rightrightarrows_R^{\phi',\sigma',k'} t^{:S} \rightrightarrows_R^{\phi'',\sigma'',k''} t''^{:S''}$, where $k' \geq k''$, $O_{(t^{:S},\phi',\sigma',k')}.\mathcal{NVO}cc(l^{:S_l}) \cap O_{(t^{:S},\phi'',\sigma'',k'')} = \emptyset$ and there is a $\omega \in O_{(t^{:S},\phi',\sigma',k')}.\mathcal{VO}cc(l^{:S_l})$, s.t. there exists a ω' with $\omega.\omega' \in O_{(t^{:S},\phi'',\sigma'',k'')}$.*

Then either $t'^{:S'} \overset{*}{\rightrightarrows}_R \circ \leftrightarrow\!\!\!\!\rightarrow_{LCP(\phi',\phi'')} \circ \overset{*}{\rightrightarrows}_R t''^{:S''}$ or $t'^{:S'} \overset{*}{\rightrightarrows}_R \circ \overset{*}{\rightrightarrows}_R t''^{:S''}$.

Proof: Let w.l.o.g. $t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_Q]_{k'}[\sigma''(g^{:S_g})]_{O_{(\phi'',\sigma'',k'')}}$,
$t'^{:S'} \cong_d t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_O]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_Q]_{k'}[\sigma''(g^{:S_g})]_{O'}$,
s.t. $O_{(t^{:S},\phi',\sigma',k')}.O \uplus O' = \bigcup_{i \geq k'} O_{(t'^{:S'},\phi'',\sigma'',i)}$ and
$t''^{:S''} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_{O''}]_{O_{(t^{:S},\phi',\sigma',k')}}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_Q]_{k'}[\sigma''(d^{:S_d})]_{O'''}[\sigma''(g^{:S_g})]_{O' \setminus O'''}$,
where $O_{(t^{:S},\phi',\sigma',k')}.O'' \cup O''' = O_{(t''^{:S''},\phi'',\sigma'',k'')}$. Hence, O' stands for the redexes of ϕ'' using σ'' outside of $\sigma'(l^{:S_l})$.

First, we can bottom-up reduce the terms $t'^{:S'}$ and $t''^{:S''}$ up to k' with (ϕ'', σ''). Let $\overline{t'^{:S'}}$ and $\overline{t''^{:S''}}$ be the result of this reduction in the left resp. right sequence. If there is a reduction at an occurrence $\omega \in O_{(t^{:S},\phi',\sigma',k')}.\mathcal{NVO}cc(l^{:S_l})$ in the right reduction sequence, then there is also a corresponding layer critical pair, giving us the convergence like in Lemma 11.24.

Otherwise, we are sure that there exists a substitution $\overline{\sigma'}$, such that $O_{(\overline{t''^{:S''}},\phi',\overline{\sigma'},k')}$ subsumes $O_{(t^{:S},\phi',\sigma',k')}$ and $\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_Q \cong_d \overline{\sigma'}(l^{:S_l})$
This case is illustrated in figure 14. Therefore,

$$\overline{t'^{:S'}} \rightrightarrows_R^{\phi',\overline{\sigma'},k'} t^{:S}[\overline{\sigma}(r^{:S_r})[\sigma''(d^{:S_d})]_{Q'}[\sigma''(g^{:S_g})]_{Q''}]_{O_{\overline{(t'^{:S'},\phi',\overline{\sigma'},k')}}} \rightrightarrows_R^{\phi',\overline{\sigma'},k'} \overline{t''^{:S''}},$$

because of $O_{\overline{(t'^{:S'},\phi',\overline{\sigma'},k')}} \uplus O_{(t^{:S},\phi',\sigma',k')} = O_{\overline{(t''^{:S''},\phi',\overline{\sigma'},k')}}$ and Lemma 11.20.

□

11.5.3 Orientation and Simplification

Lemma 11.26 *Let $t^{:S} \leftrightarrow\!\!\!\!\rightarrow_E^{p^{:S}=q^{:S'},\sigma,O} t'^{:S'}$, $(\phi : p^{:S} \to q^{:S \cup S'}) \in R$ and $(\phi' : q^{:s} \to q^{:s \cup S \setminus S'}$ if $s \not\subseteq S \setminus S') \in D$, s.t. $sort(q) \approx S \cup S'$ if $q \in \mathcal{X}_\Diamond$.*

Then $t^{:S} \overset{*}{\rightrightarrows}_R^{\phi} \circ \overset{*}{\rightrightarrows}_R^{\phi} \circ \overset{*}{\leftarrow}_D^{\phi'} t'^{:S'}$.

[2]Remark that the used substitution for ϕ'' is always σ'' !

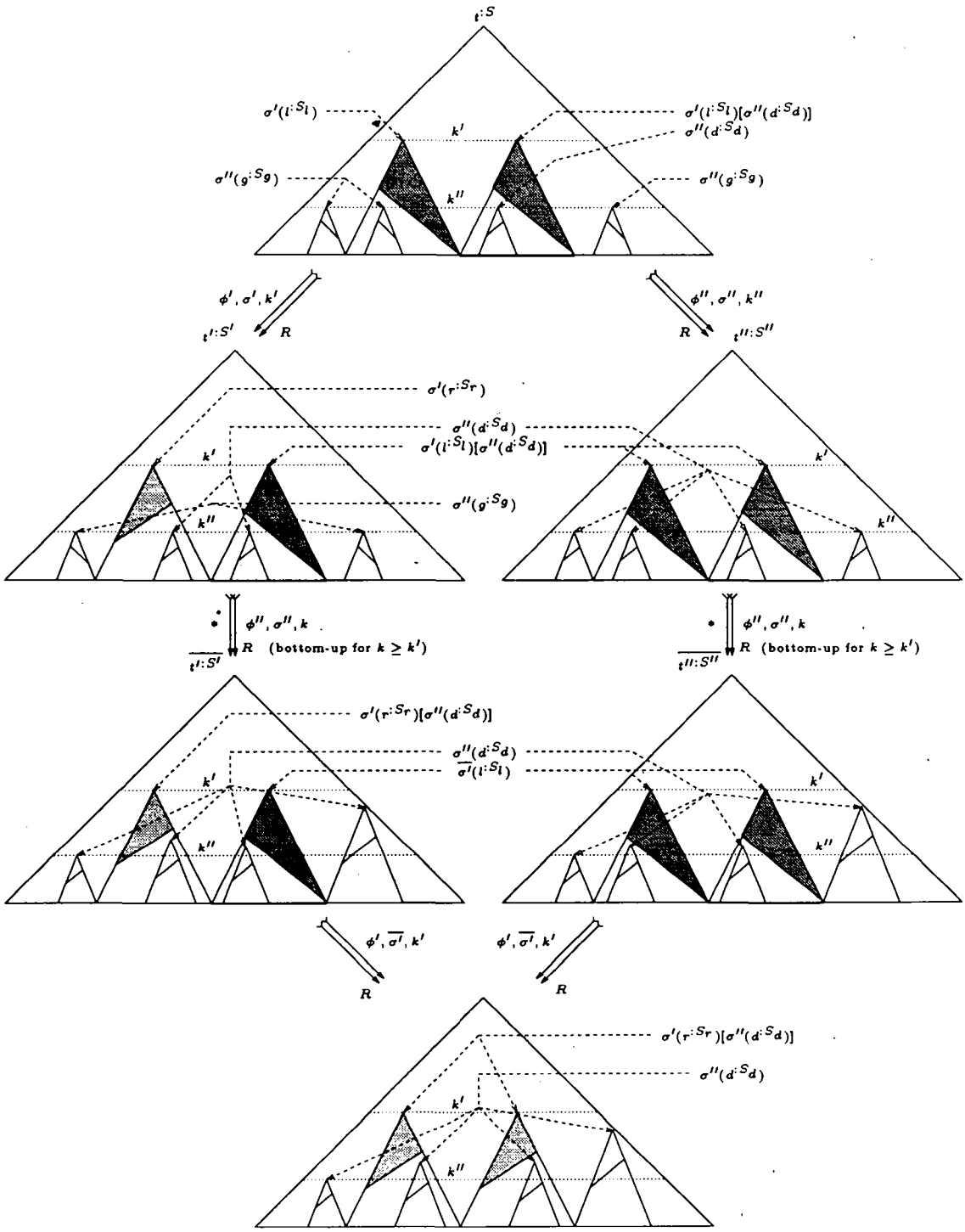


Figure 14: Variable Overlap Case, Simple Part

Proof: Let $K = \{k \mid \exists \omega \in O : |\omega| = k\}$ and $k_1, \ldots k_n$ be the elements of K sorted , s.t. $k_i > k_{i+1}$. Then $t^{:S} \rightrightarrows_R^{\phi,\sigma,k_1} \circ \ldots \circ \rightrightarrows_R^{\phi,\sigma,k_n} t''^{:S''} \leftrightarrows_R^{\phi,\sigma,k_n} \circ \ldots \circ \leftrightarrows_R^{\phi,\sigma,k_1} t'''^{:S'''} \overset{0,1}{\leftrightarrows}_D^{\phi',\sigma,O} t'^{:S'}$, where $t'^{:S'} \cong_d t^{:S}[\sigma(q^{:S'})]_O$, $t''^{:S''} \cong_d t^{:S}[\sigma(q^{:S \cup S'})]_{\overline{O}}$ and $t'''^{:S'''} \cong_d t^{:S}[\sigma(q^{:S'})^{:S \cup S'}]_O$ s.t. for all $\omega \in O_{(t^{:S},\phi,\sigma,k_i)}$, $i \in [1..n]$, there is a $\omega' \in \overline{O}$ with $\omega' \preceq \omega$. Remark that this works since bottom-up layer rewriting always yields a unique term (see Lemma 11.20).

□

The same technique of bottom-up layer rewriting is possible for the proof reduction for simplification rules using decorated rewrite rules. However, care has to be taken of the parallelism in layer rewriting, i.e. you can only simplify in layer k with (ϕ, σ) if there is no possibility to overlap the simplifying rule into the left or right hand side of the rule to be simplified (using unification instead of matching), except when the corresponding subterms are all identical and in the same layer k. Furthermore, the simplified term must not contain a new redex for (ϕ, σ), s.t. the corresponding matcher can be unified with σ.

Another possibility would be to generate several simplified decorated equations/rules for the different combinations of possible overlaps. Then, the final narrowing steps as defined for layer critical pairs are also necessary for simplification. However, this can be seen as an optimization, since we only want to prove sort inheritance and do not plan to use the resulting set of decorated and decoration rewrite rules as operational version of the G-algebra $\mathcal{P}$.

Simplification with decoration rewrite rules does not change w.r.t. standard rewriting, since their application strategy did not change. The simplification of decoration rewrite rules with decorated rewrite rules has to be restricted to layer rewriting.

11.5.4 Confluence

Let $\Rrightarrow$ be the proof reduction for layer rewriting and $\mathcal{SLC}$ be completion rules that can be easily obtained as special case of $\mathcal{OSC}$ by using layer rewriting for R instead of standard rewriting. This is justified by Lemmas 11.23, 11.25, 11.26 and 11.24, together with the remarks of the last section concerning the simplification rules. Remark that the reduction rules dealing with decoration rewrite rules can be taken from $\mathcal{OSC}$ without any changes and that all newly introduced terms at step k must be reachable from old ones using $\overset{*}{\rightarrowtail}_{D_k \cup R_k}$, as this is the case for the proof reductions in the last two sections. The proof of well-foundedness for $\Rrightarrow$, at least for the part without simplification rules, is straightforward using the complexity measure c of lemma 10.4.

In order to achieve confluence for layer rewriting, we first have to prove the conservation of the typability property for all proofs stemming from one constructed by Theorem 8.1. This is done in two steps. First, we prove the typability of all terms in decorated rewrite rules and equations. Then, we prove the typability of all terms in proofs stemming from typable proofs.

Lemma 11.27 *Let $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using $\mathcal{SLC}$. Then for all $k \geq 0$, all terms that are typable in D_k are also typable in D_∞, if D_∞ only contains flat decoration rewrite rules.*

Proof: The proof of Lemma 11.15 also works for flat decoration rewrite rules, provided they are simplified by decorated rewrite rules via layer rewriting, as supposed for $\mathcal{SLC}$. □

Lemma 11.28 *Let $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using $\mathcal{SLC}$. Then for all $k \geq 0$, all terms in CT^k_{DER} are typable in D_∞, if D_∞ only contains flat decoration rewrite rules.*

Proof: The proof is identical with the one of Lemma 11.16. Only the reference to Lemma 11.14 has to be replaced by one to 11.19. □

Theorem 11.29 *Let $\mathcal{P}_\infty \neq (\bot, \bot, \bot)$ be the presentation obtained from $(D_\mathcal{P}, E_\mathcal{P}, \emptyset)$ using $\mathcal{SLC}$, s.t. all terms in D_∞ are flat. Let furthermore $E_\infty = \emptyset$ and all critical pairs of $D_\infty \cup R_\infty$ be in $D_* \cup R_*$. Then the initial presentation $\mathcal{P}_0$ is sort inheriting on $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ and $D_\infty \cup R_\infty$ is Church-Rosser, type and existentially complete.*

Proof: The proof is once more identical to the one of section 11.4, where the references to Lemmas 11.16, 11.14 and 11.15 have to replaced by those to Lemmas 11.28, 11.19 and 11.27, respectively. □

11.6 General Term Declarations

Let us first define some needed notions and assumptions, in order to shorten Lemmas and proofs in the following. The subsumption of sets of decoration rewrite rules is a property that can be seen as combination of set subsumption combined with the rule **Subsume_deco**, i.e. D subsumes D', written $D' \subsetneq_D D$, if forall ϕ' in D', there is a ϕ in D that subsumes ϕ in the sense of **Subsume_deco** with $S \approx S'$. Let $\phi' : l^{:S_l} \to r^{:S_r}$ and $\phi'' : g^{:S_g} \to d^{:S_d}$ be decorated rewrite rules in R in the sequel.

11.6.1 Definition and Simple Properties

Working without restrictions on term declarations compels us to define a new proof reduction. Typability of terms in the transformed proof is obtained by proving that each new term comes from an old typable one by decorated rewriting in a maximally subterm sharing way.

Definition 11.30 *A term $t^{:S}$* rewrites in a maximally subterm sharing way *into $t'^{:S'}$ using a decorated rewrite rule $\phi : l^{:S_l} \to r^{:S_r}$ and a decorated substitution σ if:*

1. $t^{:S} \rightarrowtail_R^{O,\sigma,\phi} t'^{:S'}$
2. *and* $O = \{\omega \in Occ(t^{:S}) \mid t^{:S}{}_{|\omega} \cong_d \sigma(l^{:S_l})\}$.

This is written $t^{:S} \twoheadrightarrow_R^{\sigma,\phi} t'^{:S'}$. The set O of redex occurrences is written $O_{(t^{:S},\sigma,\phi)}$.

Instead of $t[u]_{O_{(t[u],\phi,\sigma)}}$ we may simply write $t[u]_{O_{(\phi,\sigma)}}$. The pair (ϕ, σ) will also be called a *radical.* Clearly, the maximality condition destroys once more stability by context and substitution.

Definition 11.31 MSS decorated critical pairs
(obtained by MSS superposition *into* decorated rules)
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $g^{:S_g} \to d^{:S_d}$ and $l^{:S_l} \to r^{:S_r}$ be rewrite rules in R with disjoint sets of variables.

The two rules overlap if there exists a position ω in the set of non-variable positions of $g^{:S_g}$, such that the decorated terms $g^{:S_g}{}_{|\omega}$ and $l^{:S_l}$ have the $\mathcal{T}$-complete, $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$-sound, non-empty set Ψ of strict decorated unifiers. Let $O = \{\omega_i\}_{i \in [1..n]}$ be the set of such positions and Ψ_i the corresponding unifiers.

Then for any combination ψ of some unifiers $\{\sigma \mid \exists i \in [1..n] : \sigma \in \Psi_i\}$ corresponding with the overlap positions $\overline{O} \subseteq O$, s.t. $\overline{O} \neq \emptyset$ and all occurrences in $\overline{O}$ are incomparable, the overlap produces the $\mathcal{T}$-MSS decorated critical pair ($\mathcal{T}$-MSSCP(R,R)) $(p^{:S_1} = q^{:S_2})$ where $q^{:S_2} =_d \psi(d^{:S_d})$ and $p^{:S_1} =_d \psi(g^{:S_g}[r^{:S_r}]_{\overline{O}})$.

Note that the computation of $\overline{O}$ can be restricted to the maximal set of redex positions in $\psi(g^{:S_g})$ for the radical $(l^{:S_l} \rightharpoonup r^{:S_r}, \psi)$, denoted by $O_{(\psi(g^{:S_g}), l^{:S_l} \to r^{:S_r}, \psi)}$. In this case, the term $p^{:S_1}$ of the MSS decorated critical pair is obtained by maximal subterm sharing rewriting from $\psi(g^{:S_g})$ with the radical $(l^{:S_l} \to r^{:S_r}, \psi)$.

Definition 11.32 MSS decoration critical pairs
(obtained by MSS superposition *into* decoration rules)
Let $\mathcal{T} \subseteq \mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$, $g^{:s} \to g^{:s \cup S_g}$ if $S_g \not\subseteq s$ *and* $l^{:S_l} \to r^{:S_r}$ *be in* D *resp.* R *with disjoint sets of variables.*

The two rules overlap if there exists a position $\omega \neq \Lambda$ *in the set of non-variable positions of* $g^{:\emptyset}$, *such that the decorated terms* $g^{:\emptyset}{}_{|_\omega}$ *and* $l^{:S_l}$ *have the* $\mathcal{T}$*-complete,* $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$*-sound, non-empty set* Ψ *of strict decorated unifiers. Let* $O = \{\omega_i\}_{i \in [1..n]}$ *be the set of such positions and* Ψ_i *the corresponding unifiers.*

Then for any combination ψ *of some unifiers* $\{\sigma \mid \exists i \in [1..n] : \sigma \in \Psi_i\}$ *corresponding with the overlap positions* $\overline{O} \subseteq O$, *s.t.* $\overline{O} \neq \emptyset$ *and all occurrences in* $\overline{O}$ *are incomparable, the overlap produces the* $\mathcal{T}$*-MSS decoration critical pair (*$\mathcal{T}$*-MSSCP(R,D))* $(p^{:s} = q^{:s \cup S}$ if $S_g \not\subseteq s)$ *where* $q^{:s \cup S} =_d \psi(g^{:s \cup S_g})$ *and* $p^{:s} =_d \psi(g[r^{:S_r}]_{\overline{O}})^{:s}$.

Let $MSSCP(R,R)_{|_\mathcal{T}}$ resp. $MSSCP(R,D)_{|_\mathcal{T}}$ denote the set of such MSS critical pairs. But in order to keep the property of typability, we also need additional decoration rules for all subterms of a MSS decoration critical pair: the set $\overline{MSSCP(R,D)}$ is defined as

$$\{(p_{|_\omega}{}^{:s} \to p_{|_\omega}{}^{:s \cup S_\omega} \text{ if } S_\omega \not\subseteq s) \mid \begin{array}{l}(p^{:s} = q^{:s \cup S} \text{ if } S_g \not\subseteq s) \in MSSCP(R,D) \\ \text{and } \omega \in \mathcal{NVO}cc(p^{:S}) \text{ and } S_\omega = Deco(p^{:S}{}_{|_\omega})\}\end{array}$$

Let us give two examples for MSSCP computations:

Example 11.33 *Let* $\begin{array}{l}(\psi : g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:s} \to g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:s \cup \{\langle B\rangle\}}) \in D, \\ (\psi' : f(a^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle A\rangle} \to y^{:\langle A\rangle}) \in R.\end{array}$
Then there is a MSS decoration critical pair $(p^{:s} = q^{:s \cup S}$ if $\{\langle A\rangle\} \not\subseteq s)$ *of* ψ' *and* ψ, *s.t.* $p^{:s} =_d g(a^{:\langle A\rangle})^{:s}$ *and* $q^{:s \cup S} =_d g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:s \cup \{\langle A\rangle\}}$.
Consequently, $\overline{MSSCP(R,D)}$ *results in:*

$$\begin{array}{l}(g(a^{:\langle A\rangle})^{:s} \to g(a^{:\langle A\rangle})^{:s \cup \{\langle A\rangle\}} \text{ if } \{\langle A\rangle\} \not\subseteq s), \\ (a^{:s} \to a^{:s \cup \{\langle A\rangle\}} \text{ if } \{\langle A\rangle\} \not\subseteq s).\end{array}$$

The second example handles the same rules as example 11.22.

Example 11.34 *Let us define two decorated rewrite rules:*
$(\psi : g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} \to g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$ *and* $(\psi' : f(z^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle} \to z^{:\langle A\rangle})$.

Clearly, ψ' *overlaps into* ψ *at occurrences* 1, 2 *and* 2.1 *with unifiers* $\sigma_1 = \{z^{:\langle A\rangle} \mapsto x^{:\langle A\rangle}\}, \sigma_2 = \{z^{:\langle A\rangle} \mapsto f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle} \mapsto a^{:\langle A\rangle}\}$, *respectively* $\sigma_{2.1} = \{z^{:\langle A\rangle} \mapsto y^{:\langle A\rangle}\}$. *Then the MSSCPs are:*

1. *overlap at* 1: $g(x^{:\langle A\rangle}, f(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$,
2. *overlap at* 2: $g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$
3. *overlap at* 2.1:
   $g(f(x^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(z^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, z^{:\langle A\rangle})^{:\langle B\rangle})$,
4. *Combining* 1 *and* 2:
   $g(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(f(y^{:\langle A\rangle}, a^{:\langle A\rangle})^{:\langle A\rangle}, y^{:\langle A\rangle})^{:\langle B\rangle})$
5. *Combining* 1 *and* 2.1:
   $g(x^{:\langle A\rangle}, f(x^{:\langle A\rangle}, y'^{:\langle A\rangle})^{:\langle A\rangle})^{:\{B\}} = g(x^{:\langle A\rangle}, x^{:\langle A\rangle})^{:\langle B\rangle})$,

The redexes 2 and 2.1 cannot be combined, since they are comparable.

In contrast to the standard case, we need to assure the typability of the term in the resulting rule by adding explicitly decoration rewrite rules. This is a consequence of the fact that $p^{:S}$ is not obtained by rewriting in $\rightarrowtail_R$. We need this more general class of critical pairs for the reduction of peaks. Remark furthermore that $MSSCP(R,R)$ critical pairs are identical with the layer critical pairs. However, we don't need any narrowing steps in the end, since we reduce maximally in parallel.

In the following, we assume (D,E,R) to be a decorated presentation, s.t. D is confluent, $\overline{MSSCP(R,D)}_{|_\mathcal{T}} \subseteq_D D$ for $\mathcal{T} = reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond,\mathcal{F},\mathcal{X}_\Diamond)^{:\downarrow\emptyset})$ and $t^{:S}$ be a bottom-up typable decorated term, as well as all terms in CT_{DER}. We call a term bottom-up typable if there is a typing proof in $\overset{*}{\rightarrowtail}_D$, that is bottom-up.

Lemma 11.35 *Let $t^{:S} \rightarrowtail_R^{\phi,\sigma} t'^{:S'}$. Then $t'^{:S'}$ is also bottom-up typable.*

Proof: For the terms in $\mathcal{I}m(\sigma)$, we can use the corresponding parts of the typing proof of $t^{:S}$, since decorated matching is subterm conservative(see Lemma 11.12). The same is true for all occurrences being incomparable with the redex positions. For the right-hand side of ϕ, we can take the instantiation of the old proof by σ. Remain the occurrences ω above the redexes. There, we can use the critical pairs from ϕ into $\phi' \in D$ in case of $O_{(t^{:S},\phi,\sigma)} \cap \omega.\mathcal{NVO}cc(lhs(\phi')) \neq \emptyset$ (critical overlap), that are assumed to be included in D, and the decoration rewrite rules for $t^{:S}$ in case of $O_{(t^{:S},\phi,\sigma)} \cap \omega.\mathcal{VO}cc(lhs(\phi')).\omega' \neq \emptyset$ for some ω' (variable overlap). In the last case, the used substitution can obviously be adapted for $t'^{:S'}$, since we reduced all identical subterms at the same time.

More formally, if Ψ_i is the bottom-up typing proof of $\sigma(x_i^{:S_i})$, where $\mathcal{D}om(\sigma) = \{x_i^{:S_i} \mid i \in [1..n]\}$, and Ψ is the bottom-up typing proof of $r^{:S_r}$, then there exists a $t''^{:S''}$, s.t.

$$t^{:\downarrow\emptyset}[\sigma(r^{:S_r})^{:\downarrow\emptyset}[(\Psi_i)_{i\in[1..n]}]_{\mathcal{VO}cc(r^{:S_r})} + \sigma(\Psi)]_{O_{(t^{:S},\phi,\sigma)}} : t'^{:\downarrow\emptyset} \overset{*}{\rightarrowtail}_D t''^{:S''}$$

and $t''^{:S''} \overset{*}{\rightarrowtail}_{\overline{MSSCP(\phi,D)}_{|_\mathcal{T}}} t'^{:S'}$ where $\mathcal{T} = reach_D(\mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond,\mathcal{F},\mathcal{X}_\Diamond)^{:\downarrow\emptyset})$ Now, every $\overset{*}{\rightarrowtail}_{D\cup\overline{CP(\phi,D)}}$ step can be replaced by some step in D, since $\overline{MSSCP(R,D)}_{|_\mathcal{T}} \subseteq_D D$ for $\mathcal{T} = reach_D(\mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond,\mathcal{F},\mathcal{X}_\Diamond)^{:\downarrow\emptyset})$ and $t^{:S}, lhs(\phi), rhs(\phi)$ are in $reach_D(\mathcal{V}alid\mathcal{T}_d(\mathcal{S}_\Diamond,\mathcal{F},\mathcal{X}_\Diamond)^{:\downarrow\emptyset})$. □

The next lemma is needed in order to prove the applicability of decoration rewrite rules or decorated equalities resulting from MSS critical pairs computation in our peak reductions.

Lemma 11.36 *Let $t^{:S}, l^{:S_l}$ be valid, typable decorated terms and σ be a decorated substitution, s.t. $\exists\omega \in \mathcal{O}cc(t^{:S}) : t^{:S}{}_{|_\omega} \cong_d \sigma(l^{:S_l})$.*

If $t^{:S} \rightarrowtail_R^{\phi'',\sigma'',O''\cup\{\omega.\omega'\}} t'^{:S'}$ with $\omega' \in \mathcal{NVO}cc(l^{:S_l})$, then

1. *$\exists\psi \in SU(g^{:S_g}{}_{|'_\omega} \cong_d^? l^{:S_l}), \exists\tau : \tau \circ \psi(t^{:S}) \cong_d \sigma''(t^{:S})$, ψ computed by* $\mathbf{UNIF_d}$ *and*
2. *$\nexists\omega'' \in \mathcal{O}cc(\psi(l^{:S_l})) \setminus \mathcal{O}cc(l^{:S_l}) : t^{:S}{}_{|_{\omega.\omega''}} \cong_d \sigma''(g^{:S_g})$.*

Proof: The schema of $t^{:S}$ is shown in Figure 15.

Let w.l.o.g. $\mathcal{V}ar(g^{:S_g}), \mathcal{V}ar(l^{:S_l}), \mathcal{V}ar(t^{:S})$ be disjoint. Hence, $\mathcal{D}om(\sigma), \mathcal{D}om(\sigma''), \mathcal{I}m(\sigma)$ and $\mathcal{I}m(\sigma'')$ are disjoint, too.

Assume that there is some x occurring at ν in $l^{:S_l}$, s.t. $\exists\nu' : \sigma''(g^{:S_g}) \cong_d \sigma(x^{:\downarrow\emptyset})_{|'_\nu}$ and $\omega'' = \nu.\nu'$ is in $\mathcal{O}cc(\psi(l^{:S_l})) \setminus \mathcal{O}cc(l^{:S_l})$.

Then, there must be some $y \in \mathcal{V}ar(l^{:S_l}) \cup \mathcal{V}ar(g^{:S_g})$ occurring at $\omega'.\upsilon$, for some $\upsilon \neq \Lambda$ with $\upsilon \in \mathcal{O}cc(g^{:S_g})$, in $\sigma(l^{:S_l})$ with $\sigma''\sigma(y^{:\downarrow\emptyset}) \cong_d \sigma(x^{:\downarrow\emptyset})$, since decorated unification is subterm conservative. Consequently, $\sigma''(g^{:S_g})_{|_\nu} = \sigma''(g^{:S_g})$ in contradiction to $\nu \neq \Lambda$. □

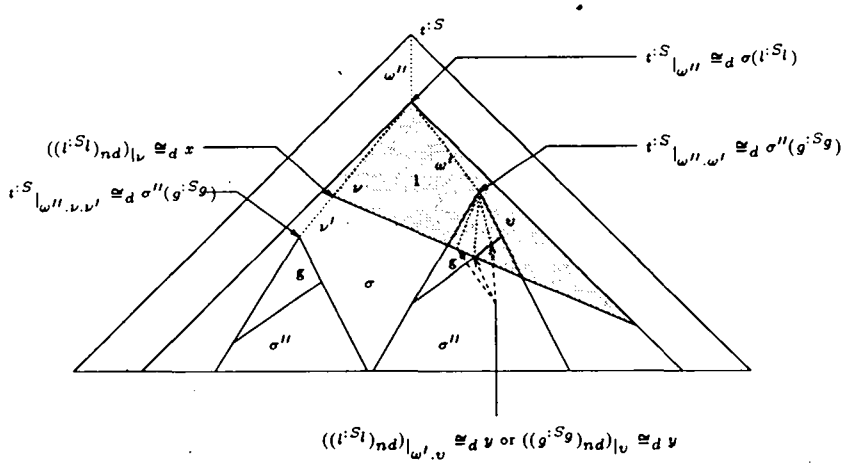


Figure 15: Term schema for applicability of MSS critical pairs

Using the same arguments as in the proof of 9.10 we get:

Corollary 11.37 *Let $t:V, t':S', t'':S'' \in \mathcal{T}$ be decorated terms such that:*

$$t':S' \leftarrow^{\omega,\alpha,\phi'}_{R} t:V \rightarrowtail^{\Lambda,\beta,\phi''}_{D} t'':S''$$

with ω being a non-variable position in the left-hand side of ϕ''. Then, either there exists a $\mathcal{T}$-MSS decoration critical pair:

$$(p:s = q:s \cup S \text{ if } S \not\subseteq s)$$

of the rule ϕ' on the rule ϕ'' at position ω, or the peak converges trivially:

$$t':S' \rightarrowtail^{\Lambda,\beta,\phi''}_{D \cup R} t_0:S_0 \leftarrow^{\omega,\alpha,\phi'}_{D \cup R} t'':S''.$$

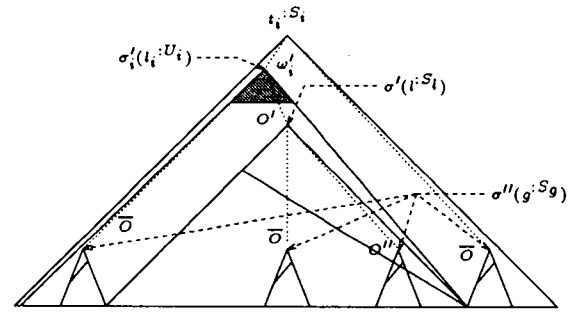
Moreover, if the peak is not trivially convergent, there is a decorated substitution ψ in a $\mathcal{T}$-complete set of decorated unifiers according to definitions 9.4, 9.5, such that $\beta\alpha \gtrsim^{W}_{d} \psi$ with $W = \mathcal{V}ar(g) \cup \mathcal{V}ar(l)$ and there exists a decorated substitution τ, such that $t':S' \cong_d \tau(p:U)$ and $t'':S'' \cong_d \tau(p:S \cup U)$ for some $U \not\subseteq S$.

Furthermore, we can extend the result of Lemma 11.35 in the following way:

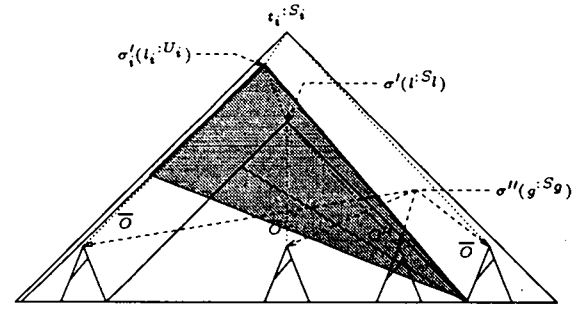
Lemma 11.38 *Let $t:S$ be bottom-up typable and $t:S \twoheadrightarrow^{\phi',\sigma'}_{R} t':S' \rightarrowtail^{\phi'',\sigma'',\overline{O}}_{R} t'':S''$, s.t.*

1. *the right-hand sides $r:S_r$ and $d:S_d$ of ϕ' and ϕ'' are bottom-up typable,*
2. *$O' = O_{(t:S,\phi',\sigma')}$,*
3. *$O'' = O_{(t:S,\phi'',\sigma'')}$,*
4. *$\overline{O} = O_{(t:S,\phi'',\sigma'')} \setminus \{\omega'' \mid \omega'' \prec O_{(t:S,\phi',\sigma')} \mathcal{NVO}cc(r:S_r)\}$ and*
5. *$O' \asymp O''$ or $\sigma''(g:S_g)$ is a subterm of $\sigma'(l:S_l)$.*

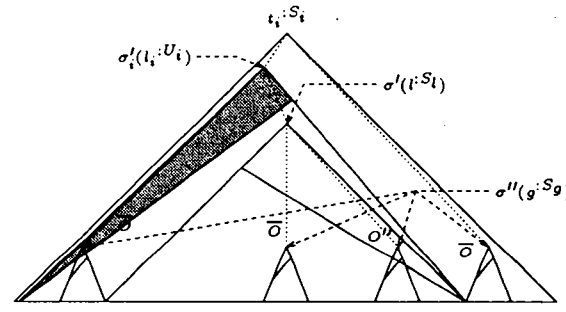
Then $t':S'$ and $t'':S''$ are also bottom-up typable.



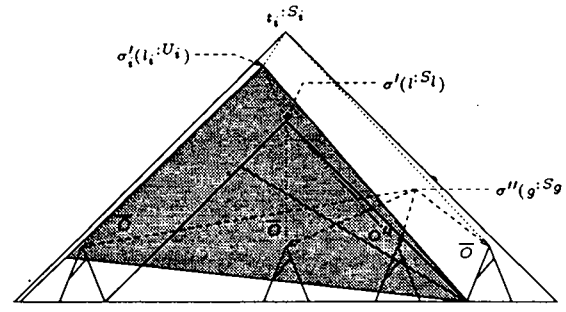
case (a) : no overlap



case (b) : overlap of ϕ' into ϕ'_i



case (c) : overlap of ϕ'' into ϕ'_i



case (d) : overlap of ϕ' and ϕ'' into ϕ'_i

Figure 16: The Four Proof Cases for Extended Type Propagation

Proof: The bottom-up typability of $t'^{:S'}$ follows immediately from Lemma 11.35. For $t''^{:S''}$ we have to construct such a proof from the one of $t^{:S}$. Remark that the case $\sigma'(l^{:S_l}) \cong_d \sigma''(g^{:S_g})$ is excluded, since this implies that $\overline{O}$ is the empty set.

Let Ψ be the assumed bottom-up typing proof for $t^{:S}$. Remember that the bottom-up typing proof Ψ' of $t'^{:S'}$ was constructed by concatenation of $\Psi_{|_{\omega \prec O'}}$, $\Psi_{|_{O'.\mathcal{NVO}cc(l^{:S_l})}}$, the bottom-up typing proof of $r^{:S_r}$ and $\overline{\Psi'}$, which is the result of decoration critical pairs computation using overlaps of ϕ' into all decoration rewrite rules that are applied above O' in Ψ.

Now, the bottom-up decoration proof Ψ'' for $t''^{:S''}$ can be constructed in almost the same way, as concatenation of $\Psi'_{|_{\omega \prec \overline{O}}}$, $\Psi'_{|_{\overline{O}.\mathcal{NVO}cc(g^{:S_g})}}$, the bottom-up typing proof of $d^{:S_d}$ and $\overline{\Psi''}$, which is defined in the following. Note that the only remaining decorations are those above the occurrences where ϕ'' was applied.

Let $\Phi' : t'^{:S'}_0 \rightarrowtail^{\phi'_1,\sigma'_1,\omega'_1}_D t'^{:S'}_1 \rightarrowtail^{\phi'_2,\sigma'_2,\omega'_2}_D \ldots \rightarrowtail^{\phi_n,\sigma_n,\omega'_n}_D t'^{:S'}_n \cong_d t^{:S}$ be $\Psi'_{|_{\omega \prec \overline{O}}}$.

Therefore we prove for all $i \in [1..n]$ the existence of ϕ''_i, σ''_i and Φ''_i corresponding with ϕ'_i, σ'_i, s.t. $\Phi''_i : t''^{:S''}_0 \rightarrowtail^{\phi''_1,\sigma''_1,\omega'_1}_D t''^{:S''}_1 \rightarrowtail^{\phi''_2,\sigma''_2,\omega'_2}_D \ldots \rightarrowtail^{\phi''_i,\sigma''_i,\omega'_i}_D t''^{:S''}_i$ provides $Deco((t''^{:S''}_i)_{|_{\omega'_i}}) \approx Deco((t'^{:S'_i})_{|_{\omega'_i}})$, $t'^{:S'}_j \rightarrowtail^{\phi'',\sigma'',\overline{O}}_R t''^{:S''}_j$ for all $j \in [0..i]$.

Therefore, we distinguish three principle cases for ω'_i and $(\phi'_i : l_i^{:s} \rightarrow l_i^{:s \cup S_i}$ if $S_i \not\subseteq s)$. The term $t_i^{:S_i}$ is illustrated in figure 16.

1. $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap \overline{O} = \emptyset$ and $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap O' = \emptyset$ (no overlap):
   We take $\phi''_i = \phi'_i$ and, assuming that $O_{\sigma''(g^{:S_g})}$ is the set of occurrences of $\sigma''(g^{:S_g})$ in $\sigma'_i(x^{:S_x})$ in $l_i^{:\emptyset}$,
   $$\sigma''_i(x^{:S_x}) \cong_d \sigma'_i(x^{:S_x})[\sigma(d^{:S_d})]_{O_{\sigma''(g^{:S_g})}}$$
   Note that the variables of $l_i^{:\emptyset}$ applied at ω'_i in $t^{:S}$ above O' are disjoint from those above $\overline{O}$, since otherwise O' were not maximal, in contradiction to its definition. Therefore, ϕ'_i is applicable with σ''_i.
2. $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap \overline{O} = \emptyset$ and $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap O' \neq \emptyset$ (ϕ' overlaps into ϕ'_i):
   Then, ϕ'_i results from a decoration critical pair between ϕ' and some decoration rule in the typing proof of $t^{:S}$. We can use the same construction as before. However, we have to be careful with the variables of $l_i^{:\emptyset}$, that were instantiated by the unifier used for the critical pairs computation.
   Assume, there were a variable x of $l_i^{:\emptyset}$ at ω''_x in $t^{:S}$ above some $\omega'' \in \overline{O}$ and parallelly at ω'''_x in $t^{:S}$ above some $\omega''' \in O'' \setminus \overline{O}$, i.e. there is some ω_g, s.t. $\omega'' = \omega'_i.\omega''_x.\omega_g$ and $\omega''' = \omega'_i.\omega'''_x.\omega_g$. If this variable existed, then ϕ'_i wouldn't be applicable anymore, since there were no matcher for x.
   Fortunately, x cannot exist, since ϕ'_i entirely contains $r^{:S_r}$ at the overlap positions and therefore all redexes for the radical (ϕ'', σ'') at such an ω''' below x have to be below $O'.\mathcal{NVO}cc(r^{:S_r})$ in $t'^{:S'}$, i.e. they are in $\overline{O}$. If ω''' is not below an overlap of ϕ' into ϕ'_i, then it must be inside or above some $\sigma'(r^{:S_r})$, that is strictly below $\omega'_i.\mathcal{NVO}cc(l_i^{:\emptyset})$. Since O' was maximal, we can be sure that the variables at ω'''_x and ω''_x are different, i.e. the two occurrences cannot exist at the same time.
3. $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap O' = \emptyset$ and $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap \overline{O} \neq \emptyset$ (ϕ'' and ϕ'_i overlap):
   Then, ϕ'_i is a rule that was already present in the typing proof of $t^{:S}$.

Furthermore, we assumed that $\sigma'(l^{:S_l})$ is no subterm of $\sigma''(g^{:S_g})$. Consequently, all variables of $l_i^{:\emptyset}$ that overlap into $\sigma''(g^{:S_g})$ are disjoint from those above $\sigma'(l^{:S_l})$, if there is a variable overlap of ϕ' into ϕ_i'. Furthermore, the MSS decoration critical pair corresponding with the overlap of ϕ'' into ϕ_i' does not instantiate the variables above $\sigma'(l^{:S_l})$, nor introduce new occurrences for them via the image of the unifier ψ. Hence, Lemma 11.36 guarantees the applicability of the decoration rewrite rule ϕ_i'' corresponding with the critical pair, using σ_i'', defined as below. Let $O_{x^{:S_x}}$ be the set of occurrences of $x^{:S_x}$ in $\psi(l_i^{:\emptyset})$ and τ be a substitution s.t. $\tau \circ \psi(t_{i-1}'^{:S_{i-1}'}) \cong_d \sigma_i' \circ \sigma''(t_{i-1}'^{:S_{i-1}'})$.

$$\sigma_i''(x^{:S_x}) = \begin{cases} \tau(x^{:S_x}) & \text{if } x^{:S_x} \in \mathcal{D}om(\psi) \\ \sigma_i'(x^{:S_x}) & \text{if } x^{:S_x} \notin \mathcal{D}om(\psi) \text{ and } t^{:S}|_{\omega_i'.O_{x^{:S_x}}} \not\trianglelefteq \sigma''(g^{:S_g}) \\ \sigma_i'(x^{:S_x})[\sigma''(g^{:S_g})]_O & \text{if } \omega_i'.O_{x^{:S_x}}.O \subseteq \overline{O} \end{cases}$$

where O has to be maximal. Remark that all variables in $\mathcal{D}om(\tau)$ have strict subterms of $\sigma''(g^{:S_g})$ as image in σ_i' or σ'', i.e. they cannot contain $\sigma''(g^{:S_g})$ as a whole in its image. Note furthermore that the last case implicitly contains the condition $x^{:S_x} \notin \mathcal{D}om(\psi)$.
Assume once more, there were a variable x in $l_i^{:\emptyset}$ at ω_x'' above some $\omega'' \in \overline{O}$ and parallelly at ω_x''' above some $\omega''' \in O'' \setminus \overline{O}$, i.e. there is some ω_g, s.t. $\omega'' = \omega_i'.\omega_x''.\omega_g$ and $\omega''' = \omega_i'.\omega_x'''.\omega_g$. Here we need the information that ϕ_i' stems from the typing proof of $t^{:S}$. The redex $\sigma''(g^{:S_g})$ at ω''' in $t'^{:S'}$ has to be above or below an occurrence of $\sigma'(r^{:S_r})$ stemming from a maximally subterm sharing decorated rewriting step. As already mentioned, ϕ_i' was already used in the typing proof for $t^{:S}$ and therefore all variables in $l_i^{:\emptyset}$ above some $\sigma'(l^{:S_l})$ have to be disjoint from those above some occurrence in O'' incomparable to all occurrences in O' - otherwise O' were not maximal. Consequently, ω'' and ω''' cannot exist at the same time.

4. $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap O' \neq \emptyset$ and $\omega_i.\mathcal{NVO}cc(l_i^{:\emptyset}) \cap \overline{O} \neq \emptyset$ (ϕ' and ϕ'' overlap into ϕ_i'):
Then, there is a MSS decoration critical pair corresponding with the overlap of ϕ'' into ϕ_i'. Let ψ be used the unifier. Then the decoration rule ψ_i'' resulting from the decoration critical pair can be applied using the decorated substitution σ_i'' defined as in the last case, where τ defined by $\tau \circ \psi(t_{i-1}'^{:S_{i-1}'}) \cong_d \sigma_i' \circ \sigma''(t_{i-1}'^{:S_{i-1}'})$, using Lemma 11.36. The consistency of the variable images also follows from the arguments used in the last two cases.

□

11.6.2 Peak Reduction

The reduction of peaks between two decoration rule applications and between a decoration rewrite rule and a decorated rewrite rule are as usual. Note that the peaks corresponding with critical pairs in $MSSCP(R,D)$ differ slightly, due to the multiple occurrences decorated rewriting relation, but the application of a decoration rewrite rule strictly above a decorated rewrite rule ϕ using σ cannot introduce a new redex for the radical (ϕ,σ). Therefore, the proof reduction remains essentially the same. However, the peaks between two decorated rewrite rules applications are reduced in a different way.

As in the last section, we assume (D,E,R) to be a decorated presentation, s.t. D is confluent, $\overline{MSSCP(R,D)}|_{\mathcal{T}} \subseteq_D D$ for $\mathcal{T} = reach_D(\mathcal{T}_d(\mathcal{S}_\diamond,\mathcal{F},\mathcal{X}_\diamond)^{:\emptyset})$ and $t^{:S}$ be a bottom-up typable decorated term, as well as all terms in CT_{DER}.

The first case is where there are disjoint sets of redexes for the radicals (ϕ',σ') and (ϕ'',σ'').

Lemma 11.39 *Let* $t'^{:S'} \leftarrow_R^{\phi',\sigma',O'} t^{:S} \rightarrow_R^{\phi'',\sigma'',O''} t''^{:S''}$, *s.t.* $O_{(t^{:S},\phi',\sigma')} \asymp O_{(t^{:S},\phi'',\sigma'')}$.
Then there exists $\Psi : t'^{:S'} \overset{0,1}{\mapsto}_R \circ \rightarrowtail_R \circ \leftarrowtail_R \circ \overset{0,1}{\leftarrow}_R t''^{:S''}$, *s.t. all new terms are typable.*

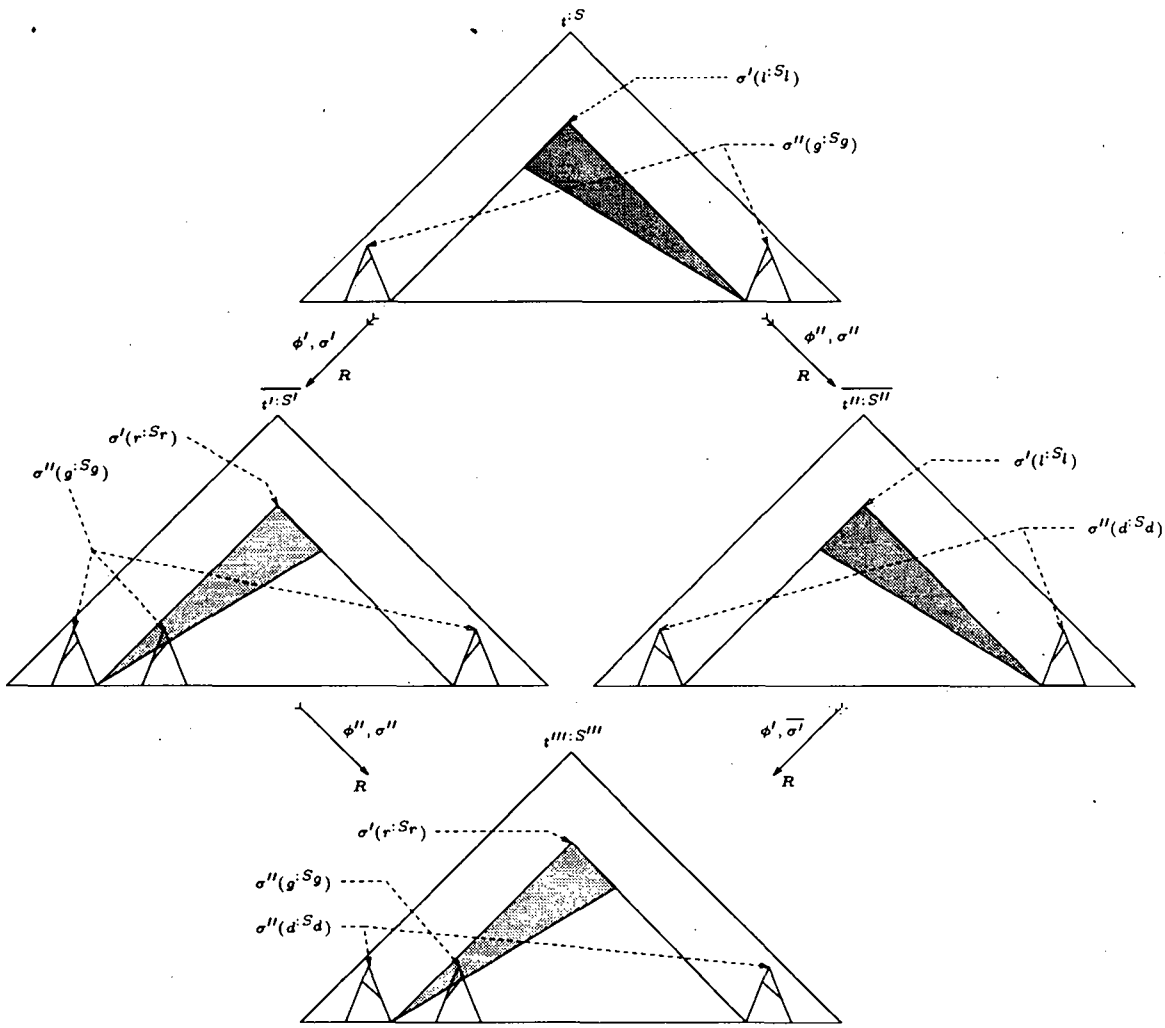


Figure 17: No Overlap Case

Proof: Let $\overline{t'^{:S'}}, \overline{t''^{:S''}}$ be decorated terms, s.t. $\overline{t'^{:S'}} \hookleftarrow_R^{\phi',\sigma'} t^{:S} \hookrightarrow_R^{\phi'',\sigma''} \overline{t''^{:S''}}$. Therefore, $t'^{:S'} \overset{0,1}{\hookrightarrow}_R^{\phi',\sigma',\overline{O'}} \overline{t'^{:S'}}$ and $t''^{:S''} \overset{0,1}{\hookrightarrow}_R^{\phi'',\sigma'',\overline{O''}} \overline{t''^{:S''}}$, where $O' \cup \overline{O'} = O_{(t^{:S},\phi',\sigma')}$ and $O'' \cup \overline{O''} = O_{(t^{:S},\phi'',\sigma'')}$. The resulting peak is illustrated in figure 17.

Let furthermore $t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(\phi',\sigma')}}[\sigma''(g^{:S_g})]_{O_{(\phi'',\sigma'')}}$,

i.e. $\overline{t'^{:S'}} \cong_d t^{:S}[\sigma'(r^{:S_r})]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(g^{:S_g})]_{O_{(t^{:S},\phi'',\sigma'')}}$

and $\overline{t''^{:S''}} \cong_d t^{:S}[\sigma'(l^{:S_l})]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(d^{:S_d})]_{O_{(t^{:S},\phi'',\sigma'')}}$,

since $\sigma''(g^{:S_g})$ cannot contain $\sigma'(l^{:S_l})$ and vice versa.

First of all, remark that $\overline{t'^{:S'}}$ and $\overline{t''^{:S''}}$ are bottom-up typable by Lemma 11.35. Now, we can reach convergence of the peak:

$$\overline{t'^{:S'}} \rightarrowtail_R^{\phi'',\sigma'',O_{(t^{:S},\phi'',\sigma'')}} t'''^{:S'''} \leftarrowtail_R^{\phi',\sigma',O_{(t^{:S},\phi',\sigma')}} \overline{t''^{:S''}}$$

The typability of $t'''^{:S'''}$ can be obtained by Lemma 11.38. □

In the second case, there are only variable overlaps and no non-variable overlaps.

Lemma 11.40 *Let* $t'^{:S'} \leftarrowtail_R^{\phi',\sigma',O'} t^{:S} \rightarrowtail_R^{\phi'',\sigma'',O''} t''^{:S''}$, *where* $O_{(t^{:S},\phi',\sigma')}.\mathcal{NVO}cc(l^{:S_l}) \cap O_{(t^{:S},\phi'',\sigma'')} = \emptyset$ *and there is a* $\omega \in O_{(t^{:S},\phi',\sigma')}.\mathcal{VO}cc(l^{:S_l})$, *s.t. there exists a* ω' *with* $\omega.\omega' \in O_{(t^{:S},\phi'',\sigma'')}$.

Then $t'^{:S'} \overset{0,1}{\hookrightarrow}_R \circ \rightarrowtail_R \circ \leftarrowtail_R \circ \overset{0,1}{\hookleftarrow}_R t''^{:S''}$, *where all new terms are typable.*

Proof: Let $\overline{t'^{:S'}}, \overline{t''^{:S''}}$ be defined as in the proof of Lemma 11.39. The resulting peak is illustrated in figure 18. The typability of $\overline{t'^{:S'}}$ and $\overline{t''^{:S''}}$ follows from Lemma 11.35. Hence, we can assume w.l.o.g.

$t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_O]_{O_{(\phi',\sigma')}}[\sigma''(g^{:S_g})]_{\overline{O}}$,

$\overline{t'^{:S'}} \cong_d t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_{O^v}[\sigma''(g^{:S_g})]_{O^{nv}}]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(g^{:S_g})]_{\overline{O}}$,

$\overline{t''^{:S''}} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(d^{:S_d})]_{\overline{O}}$.

where $O_{(t^{:S},\phi',\sigma')}.O \uplus \overline{O} = O_{(t^{:S},\phi'',\sigma'')}$, O^{nv} stands for the redexes introduced by the radical (ϕ',σ'), which overlap into non-variable positions of $\sigma'(r^{:S_r})$, and O^v for those under variable positions.

Note that all redexes at $O_{(t^{:S},\phi',\sigma')}.O^v$ in $\overline{t'^{:S'}}$ correspond with some redex at $O_{(t^{:S},\phi',\sigma')}.O$ in $t^{:S}$, since $\mathcal{V}ar(l^{:S_l}) \supseteq \mathcal{V}ar(r^{:S_r})$. The peak converges with:

$$\overline{t'^{:S'}} \rightarrowtail_R^{\phi'',\sigma'',O_{(t^{:S},\phi',\sigma')}.O^v \cup \overline{O}} t'''^{:S'''} \leftarrowtail_R^{\phi',\overline{\sigma'},O_{(t^{:S},\phi',\sigma')}} \overline{t''^{:S''}},$$

where $\overline{\sigma'}$ is defined as follows, provided $O_{\sigma''(g^{:S_g})}$ is the set of occurrences of $\sigma''(g^{:S_g})$ in $x^{:S}$:

$$\overline{\sigma'(x^{:S})} = \sigma'(x^{:S})[\sigma''(d^{:S_d})]_{O_{\sigma''(g^{:S_g})}}$$

The bottom-up typability of $\overline{t'^{:S'}}, \overline{t''^{:S''}}$ follows from the one of $t^{:S}$ together with Lemma 11.35. For the typability of $t'''^{:S'''}$ we can still use Lemma 11.38. □

The last case covers peaks where there is at least one non-variable overlap (and possibly other variable overlaps).

Lemma 11.41 MSS Critical Pair Lemma
Let $t'^{:S'} \leftarrowtail_R^{\phi',\sigma',O'} t^{:S} \rightarrowtail_R^{\phi'',\sigma'',O''} t''^{:S''}$, *s.t.* $O_{(t^{:S},\phi',\sigma')}.\mathcal{NVO}cc(l^{:S_l}) \cap O_{(t^{:S},\phi'',\sigma'')} \neq \emptyset$.

Then $t'^{:S'} \overset{0,1}{\hookrightarrow}_R \circ \overset{0,1}{\hookrightarrow}_R \circ \leftarrowtail\!\!\!\rightarrowtail_{MSSCP(\phi',\phi'')|_{\mathcal{T}}} \circ \overset{0,1}{\hookleftarrow}_R t''^{:S''}$, *where* $\mathcal{T} = reach_D(\mathcal{T}_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)^{:\mathcal{V}})$ *and all new terms are bottom-up typable.*

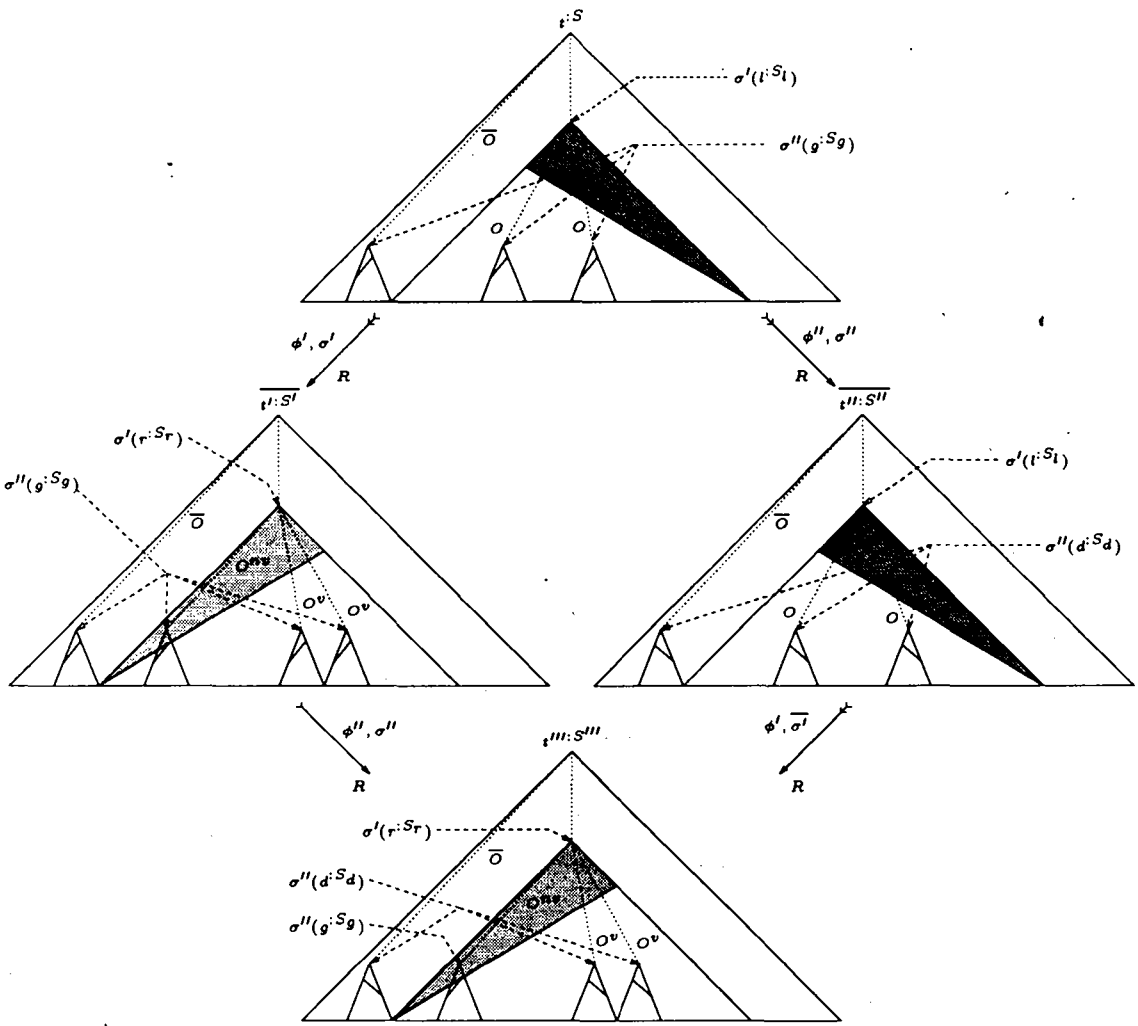


Figure 18: Variable Overlap Case

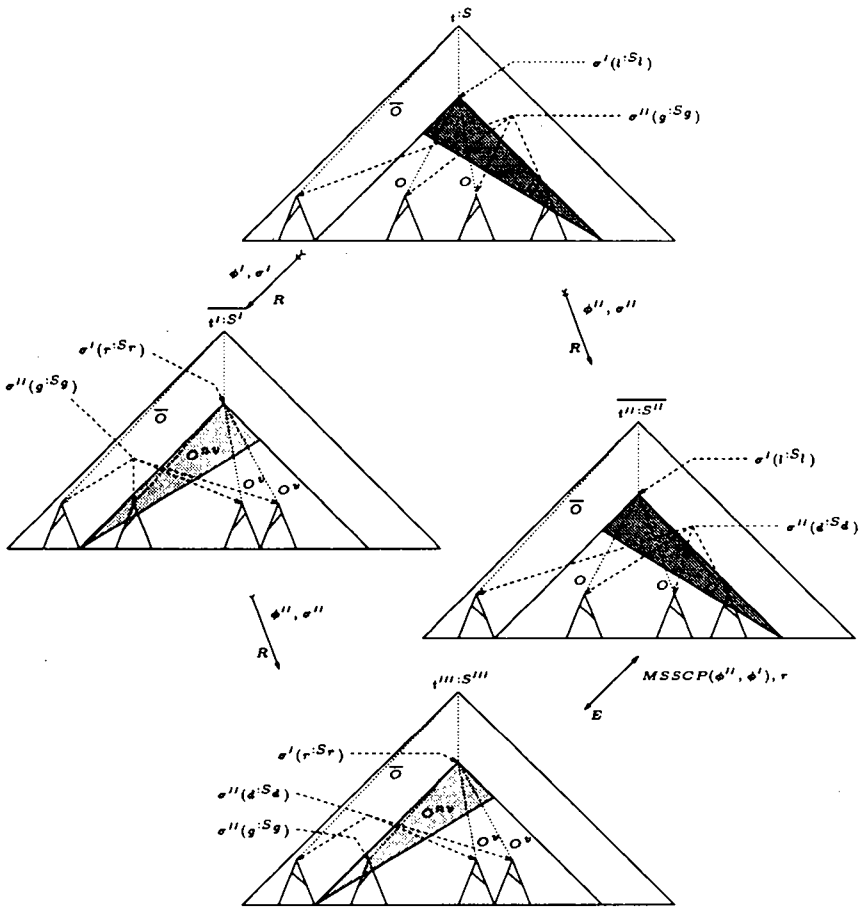


Figure 19: Critical Overlap Case

Proof: Let once more $\overline{t'^{:S'}}, \overline{t''^{:S''}}$ be defined as in the proof of Lemma 11.39.The resulting peak is illustrated in Figure 19. The typability of $\overline{t'^{:S'}}$ and $\overline{t''^{:S''}}$ follows from Lemma 11.35. We can assume w.l.o.g.

$$t^{:S} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(g^{:S_g})]_O]_{O_{(\phi',\sigma')}}[\sigma''(g^{:S_g})]_{\overline{O}},$$
$$\overline{t'^{:S'}} \cong_d t^{:S}[\sigma'(r^{:S_r})[\sigma''(g^{:S_g})]_{O^v}[\sigma''(g^{:S_g})]_{O^{nv}}]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(g^{:S_g})]_{\overline{O}} \text{ and}$$
$$\overline{t''^{:S''}} \cong_d t^{:S}[\sigma'(l^{:S_l})[\sigma''(d^{:S_d})]_O]_{O_{(t^{:S},\phi',\sigma')}}[\sigma''(d^{:S_d})]_{\overline{O}}, \text{ s.t.}$$

1. $O^v = \{\omega \mid \exists \omega', \omega'' : \omega \succeq \omega''$ with
   (a) $\omega' \in O_{(t^{:S},\phi',\sigma')}$,
   (b) $\omega'' \in \mathcal{VO}cc(r^{:S_r})$ and
   (c) $\omega'.\omega \in O_{(t^{:S},\phi'',\sigma'')}\}$,
2. $O^{nv} = O_{(\sigma'(r^{:S_r}),\phi'',\sigma'')} \setminus (O^v \cup \overline{O})$,
3. $O_{(t^{:S},\phi',\sigma')}.O \cup \overline{O} = O_{(t^{:S},\phi'',\sigma'')}$.

Remark that $O^v \cup \overline{O} \neq O_{(\overline{t'^{:S'}},\phi'',\sigma'')}$, since there might be redexes for the radical (ϕ'', σ'') above $O_{(t^{:S},\phi',\sigma')}.\mathcal{VO}cc(r^{:S_r})$, i.e. possibly $O^{nv} \neq \emptyset$. Consequently, we can let the peak converge as follows:

$$\overline{t'^{:S'}} \underset{R}{\overset{0,1}{\rightarrowtail}}{}^{\phi',\sigma',O_{(t^{:S},\phi',\sigma')}.O^v \cup \overline{O}} t'''^{:S'''} \underset{MSSCP(\phi',\phi'')_{|_T}}{\overset{\tau',O_{(t^{:S},\phi',\sigma')}}{\longleftrightarrow}} \overline{t''^{:S''}},$$

where τ' is defined via the unifier ψ used for the critical pairs computation. Let τ be defined by $\tau \circ \psi(t^{:S}) \cong_d \sigma' \circ \sigma''(t^{:S})$. Then τ' is defined as follows, provided that $O_{\sigma''(g^{:S_g})}$ is the set of occurrences of $\sigma''(g^{:S_g})$ in $x^{:S}$:

$$\tau'(x^{:S}) = \tau(x^{:S})[\sigma''(d^{:S_d})]_{O_{\sigma''(g^{:S_g})}}.$$

The first reduction with ϕ' is superfluous in the case $\sigma'(l^{:S_l}) \cong_d \sigma''(g^{:S_g})$. Lemma 11.36 guarantees that the critical pair is actually applicable. The typing proof of $t'''^{:S'''}$ can be obtained from Lemma 11.38. □

11.6.3 Orientation and Simplification

Lemma 11.42 *Let* $t^{:S} \underset{E}{\overset{p^{:S}=q^{:S'},\sigma,O}{\longleftrightarrow}} t'^{:S'}$, $(\phi : p^{:S} \to q^{:S \cup S'}) \in R$ *and* $(\phi' : q^{:s} \to q^{:s \cup S \setminus S'}$ *if* $s \not\subseteq S \setminus S') \in D$, *s.t.* $sort(q) \approx S \cup S'$ *if* $q \in \mathcal{X}_\delta$.
Then $t^{:S} \underset{R}{\overset{\phi,\sigma,O}{\rightarrowtail}} \circ \underset{D}{\overset{0,1}{\leftarrowtail}}{}^{\phi',\sigma,O} t'^{:S'}$.

Proof: The proof reduction is identical with the one for $\Longrightarrow$. □

Concerning the simplification rules for decorated rewrite rules and equalities using $\phi \in R$, it is easy to find proof reduction rules if restrictions similar to the case of layer rewriting are applied. We only need to assure, that the simplifying rule does not overlap at any other position in the left or right-hand side of the term to be reduced, except when all unifiers are identical with the matcher used for simplification. As for layer rewriting, it is possible to define simplification rules treating the different possible combinations of substitutions, yielding several decorated equalities and rewrite rules as replacement of the simplified object, but we won't go into details here. Anyhow, a type conservative proof reduction for these simplification rules may be found in the case of **Compose_R** and **Collapse_R**. For **Simplify_R**, this seems to be impossible, since there is no way to propagate typing information by an equation, if we don't want to define critical overlaps of E into D, what is definitely not the case for us.

11.6.4 Confluence

Let $\mathcal{MSSC}$ be the set of completion rules shown in Figure 20 plus **Delete**, **Orient_SD**, **Orient_NSD** of $\mathcal{OSC}$ and the failure rules **Detect**, **Extend**, that do not change. To guarantee bottom-up typability of terms, additional strategy assumptions on the completion process are needed. We call *first-level (completion) rules* those completion rules that are dealing with decorated equalities and rewrite rules only.

General Assumption 11.43 *We suppose for the strategy of $\mathcal{MSSC}$ the following points:*

1. *Sort inheritance on typable terms is tested whenever no more critical pairs in $CP(D, D)$ can be computed.*
2. *Critical pairs in $MSSCP(R, D)$ are only computed when D is confluent.*
3. *First-level completion rules are only used if no other rules apply.*
4. *Decoration rules are not composed.*
5. *Subsumption of decoration rules is tested in a strict way (i.e. $S \approx S'$ in* **Subsume_deco***).*
6. *If a decoration rewrite rule for a critical pair in $MSSCP(R, D)$ is added, then all the corresponding subterm decoration rewrite rules are added simultaneously.*

The requirements of assumption 11.43 can be fulfilled by the strategy shown in figure 21, using the syntax of ELAN [KKV93].

We clarify our syntax before starting with the propositions. The first level completion rules in $\mathcal{MSSC}$ are limited to **Deduce_MSS** for $MSSCP(R, R)$. The simplification rules will be followed by $\_D$, since $\phi \in R$ is not allowed. Furthermore, **Deduce_MSS_RR** stands for the case $MSSCP(R, R)$ and **Deduce_MSS_DR** consequently for $CP(D, R)$. Analogously, **Deduce_deco_MSS** with suffix **_DD** stands for the case $CP(D, D)$ and with suffix **_RD** for the case $MSSCP(R, D)$.

Lemma 11.44 *Let $(D_\mathcal{P}, E_\mathcal{P}, \emptyset) = (D_0, E_0, R_0) \vdash (D_1, E_1, R_1) \vdash \ldots$ be a derivation using $\mathcal{MSSC}$, satisfying assumption 11.43. Then for all $k \geq 0$, all terms in CT^k_{DER} are typable in D_k.*

Proof: The proof is an induction over the number of completion steps k. If $k = 0$, then all $t^{:S} \in CT^0_{DER}$ are equal to $t^{:\lfloor\emptyset}$ resp. $(t^{:\lfloor\emptyset})^{:S}$ for terms in decoration rewrite rules and therefore trivially typable.

If $k > 0$, then we distinguish different cases for the last applied completion rule:

1. **Deduce_deco_MSS:**
   In the case of $CP(D, D)$, the typing proofs are transformed and the new term is typable by Lemma 11.13, since it is only the instantiation of a term in CT^{k-1}_{DER}, that has a decoration enriched via $\rightarrowtail_{D_{k-1}}$.
   In the case of $CP(R, D)$, the new terms added are trivially typable, since assumption 11.43 guarantees the presence of all needed decoration rewrite rules.
2. **Subsume_deco_MSS:**
   $CT^k_{DER} = CT^{k-1}_{DER}$ and the typing proofs are only transformed.
3. **Simplify_MSS_D, Compose_MSS_D, Collapse_MSS_D:**
   Every $p^{:S} \in CT^k_{DER} \setminus CT^{k-1}_{DER}$ stems from a $p'^{:S'}$ in CT^{k-1}_{ER} via $\rightarrowtail^{\phi,\sigma,\omega}_D$, i.e. the typing proof of $p^{:S}$ is the one of $p'^{:S'}$ followed by the decoration rewriting step $\rightarrowtail^{\phi,\sigma,\omega}_D$.

1. **Deduce_deco_MSS**
$$\frac{D, E, R}{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } S \not\subseteq s)\}, E, R} \quad \text{if } (p^{:s} = p^{:s \cup S} \text{ if } S \not\subseteq s) \in \frac{}{MSSCP(R, D) \cup CP(D, D)}$$

2. **Subsume_deco_MSS**
$$\frac{D \cup \{(p^{:s} \to p^{:s \cup S} \text{ if } c(s))\}, E, R}{D, E, R} \quad \text{if } p^{:\emptyset} \rightarrowtail_D^{\Lambda, \sigma, \phi} p^{:S} \text{ and } (p^{:s} \to p^{:s \cup S} \text{ if } c(s)) \neq \phi$$

3. **Deduce_MSS**
$$\frac{D, E, R}{D, E \cup \{(p^{:S} = q^{:S'})\}, R} \quad \text{if } (p^{:S} = q^{:S'}) \in MSSCP(R, R) \cup CP(D, R)$$

4. **Simplify_MSS_D**
$$\frac{D, E \cup \{(p^{:S} = q^{:S'})\}, R}{D, E \cup \{(p''^{:S''} = q^{:S'})\}, R} \quad \text{if } p^{:S} \rightarrowtail_D^{\sigma, \phi} p''^{:S''}$$

5. **Compose_MSS_D**
$$\frac{D, E, R \cup \{(l^{:S_l} \to r^{:S_r})\}}{D, E, R \cup \{(l^{:S_l} \to r'^{:S_{r'}})\}} \quad \text{if } r^{:S_r} \rightarrowtail_D^{\sigma, \phi} r'^{:S_{r'}}$$

6. **Collapse_MSS**
$$\frac{D, E, R \cup \{(l^{:S_l} \to r^{:S_r})\}}{D, E \cup \{(l'^{:S_{l'}} = r^{:S_r})\}, R} \quad \text{if } l^{:S_l} \rightarrowtail_D^{\sigma, \phi} l'^{:S_{l'}} \;\&\; l^{:S_l} \to r^{:S_r} \gg \phi$$

Figure 20: $\mathcal{MSSC}$: Completion Rules for Maximally Subterm Sharing Rewriting.

4. **Delete, Orient_SD:**
$CT^k_{DER} \subset CT^{k-1}_{DER}$.
5. **Orient_NSD:**
Like **Simplify_D**.
6. **Deduce_MSS_DR, Deduce_MSS_RR:**
In both cases, the new terms are either an instantiation of a term in CT^{k-1}_{DER} or a reduced instantiation. In either case, the typability of the instantiation follows from Lemma 11.13 and the fact that the unifiers are calculated with $\mathbf{UNIF_d}$. If the instantiation is reduced via $\rightarrowtail_D$, then the typability follows immediately from the one of the instantiation.
If it is reduced via $\rightarrowtail_R$, then its typability is a consequence of Lemma 11.35. Remark that D_{k-1} must be confluent on all typable terms, since **Deduce_deco_MSS** is no more applicable, due to assumption 11.43, implying also $MSSCP(R_{k-1}, D_{k-1})|_{reach_{D_{k-1}}(T_d(S_0, \mathcal{F}, \mathcal{X}_0)^{:\downarrow\theta})} \subseteq_D D_{k-1}$. Furthermore, the term to be reduced must be bottom-up typable in D_{k-1}, since D_{k-1} is confluent and we do not compose or simplify decoration rewrite rules.

□

Now, we can define a typing proof preserving proof reduction $\Rrightarrow$ according to the peak reduction Lemmas. The set of reduction rules can be found in appendix A.2.

Lemma 11.45 *The proof reduction $\Rrightarrow$ is well-founded.*

```
strategy Deco_Norm
   repeat dont know choose(Deduce_MSS_DD; Subsume_deco_MSS)
   endrepeat
   dont know choose(Detect; Extend)
end of strategy

strategy Weakening
   repeat
      repeat dont know choose(Simplify_MSS_D; Compose_MSS_D; Collapse_MSS_D)
      endrepeat,
      try(Deduce_MSS_DR),
      repeat   try(Delete) endrepeat,
   endrepeat
end of strategy

strategy Propagate
   repeat
      Deco_Norm, Weakening,
      try(Deduce_MSS_RD),
   endrepeat

strategy DecoratedRewrite
   dont know choose(Deduce_MSS_RR; Orient_NSD; Orient_SD),
   try(Extend, Detect)
end of strategy

strategy MSSC
   repeat dont know choose(Deco_Norm; (Deco_Norm, Weakening);
               Propagate; (Propagate, DecoratedRewrite))
   endrepeat
end of strategy
```

Figure 21: A Sample MSSC Strategy

Proof: This can be verified with the complexity measure c of lemma 10.4. □

Theorem 11.46 *Let $\mathcal{P}_\infty \neq (\bot, \bot, \bot)$ be the presentation obtained from $(D_\mathcal{P}, E_\mathcal{P}, \emptyset)$ using $\mathcal{MSSC}$, s.t. assumption 11.43 is fulfilled. Let furthermore $E_\infty = \emptyset$ and all critical pairs of $D_\infty \cup R_\infty$ be in $D_* \cup R_*$. Then the initial presentation $\mathcal{P}_0$ is sort inheriting on $ValidT_d(\mathcal{S}_\Diamond, \mathcal{F}, \mathcal{X}_\Diamond)$ and $D_\infty \cup R_\infty$ is Church-Rosser, type and existentially complete.*

Proof: All peak reductions have the property, that they only introduce typable new terms, provided $\overline{MSSCP(R,D)} \subseteq_D D$ and the term at the top of the peak is bottom-up typable if it is a peak between two decorated rewrite rules. Bottom-up typability results from typability at each step k, when D_k is confluent on all typable terms, since all terms in the proof to be transformed are typable by induction hypothesis and we do not simplify nor compose decoration rewrite rules (see Lemma 11.7). Hence, peaks between decorated rewrite rules with variable overlap or without overlap can be reduced at any such k if additionally $\overline{MSSCP(R,D)} \subseteq_D D$ holds,

i.e. especially in D_∞. Peaks with critical overlap between decorated rewrite rules are reduced when the corresponding decorated critical pairs $MSSCP(R,R)$ are calculated. Here we can be sure, due to assumption 11.43, that all decoration critical pairs are calculated and that $\overline{MSSCP(R,D)} \subseteq_D D$, giving us the needed confluence of D_k.

If we orient a rule, the only new term can be reached via decoration rewriting from an old one, i.e. all terms in the new proof are trivially typable. Deleting an equation leads to a proof with less terms and preserves therefore also typability. All other proof reductions have the property, that any new term can be reached from an old one by decoration rewrite steps. Consequently, typability of these terms follows as in the orientation case.

Now that we know the typability of all terms in proofs in $\mathcal{P}_\infty$, we can reach sort inheritance, the Church-Rosser property, type and existential completeness as in the proof of Theorem 11.17. □

11.7 Comparison

Each of the three cases treated in the last sections offer several advantages and drawbacks:

1. The standard rewriting technique gives us the compatibility with classical order-sorted completion procedures as described in [GKK90]. Furthermore, the sort inheritance test can be postponed until the end of completion.

2. The layer rewriting technique gives us a similar property, but simplification and critical pairs computation got quite complicated.

3. The maximally subterm sharing rewriting technique has pure proof purposes and does not support simplification for decorated rewrite rules for the moment.

4. Anyway, if the proof of sort inheritance succeeds with one of these techniques, the resulting decorated presentation may be used in order to continue with $\mathcal{OSC}$, since all deductions were correct as a consequence of the fact that any layer rewriting or maximally subterm sharing step can be simulated by standard decorated rewriting.

5. We conjecture that the three techniques can be combined into one strategy, where standard rewriting is used as long as all terms in decoration rewrite rules are flat and linear, layer rewriting for the flat case and parallel rewriting on occurrence sets as long as there are non-flat decoration rules. However, Sort inheriting on typable terms must be tested each time when non-flat decoration rules are introduced.

   This seems to be correct, since all terms in rules and equations stay typable as well as all terms in the transformed proofs. Furthermore, each step in a rewriting strategy can be simulated by standard rewriting sequences or one parallel rewriting step at multiple occurrences. A standard rewriting or parallel rewriting step can be replaced by two converging bottom-up layer rewriting sequences, due to Lemma 11.20. Remark also, that we can mix the proof transformations, since their termination is provable with the same complexity measure.

Further extensions of these techniques might be the definition of simplification rules for maximally subterm sharing rewriting and an extension of the results on layer rewriting to non-flat, semi-linear terms in decoration rules, where non-linear variables may only occur in identical subterms.

11.8 Changing the Membership Relation

The application of the **Detect** rule indicates the detection of a counter-example for the sort inheritance w.r.t. $\leq_{\mathcal{S}}^{syn}$, i.e. there exists a term belonging to two incomparable sorts A, B without a common subsort. Therefore the intersection of A and B, which must be computed for the unification of two variables of sorts A and B respectively, cannot be described as a sort.

Assume that the valid decorated term $t^{:S}$ with A, B as before, is a detected counter-example for the sort inheritance of $\mathcal{P}$ w.r.t. $\leq_{\mathcal{S}}^{syn}$. Then the definition of models in G-algebra forces the interpretation of this term to be in both denotations of A and B. Hence, we can add, in a model theoretically conservative way(i.e. each model of the initial presentation can be extended to one of the new presentation and each one of the new can be reduced to one of the initial presentation), a new sort $C \notin \mathcal{S}$ and the membership formula $(t' : C)$, s.t. $t' = \alpha(t^{:S})_{nd}$ for some $\mathcal{T}(\Sigma, \mathcal{X})$-assignment α. As in the case of adding a new subsort, we have to restart the completion.

Results analogous to Proposition 10.10 for the detection of differences between $\leq_{\mathcal{S}}^{syn}$ and $\leq_{\mathcal{S}}^{sem}$ are Theorems 11.17, 11.29 and 11.46. Remark that there is no condition like $\leq_{\mathcal{S}}^{syn} = \leq_{\mathcal{S}}^{sem}$ and therefore we can be sure that $\mathcal{P}_\infty = (\bot, \bot, \bot)$ if $\leq_{\mathcal{S}}^{syn} \neq \leq_{\mathcal{S}}^{sem}$ and $\mathcal{P}$ is not sort inheriting at the same time.

12 Related Work

We first come back in this section on the relation between sort inheritance and regularity, then we compare our own completion approach with several others. First of all, we prove that the completion presented here subsumes the completion described in [GKK90] for OBJ-3 specifications. Next we compare with the tree automata approach of [Com92]. Furthermore we discuss the relation with the signature extension approach [CH91], the T-contact method [Wer93] and the approaches of L. With [Wit92], then of P. Watson and J. Dick [WD89].

12.1 Sort Inheritance vs. Regularity

Sort inheriting can be interpreted as an extension of regularity, which means that each valid term has a unique least sort. The reason why we need something like regularity is hidden in the unification used for critical pair calculation.

When we try to resolve a peak in a proof based on rewriting steps, we need to unify two rules left-hand sides and introduce a new equality based on the right-hand sides of the two rules instantiated by the unifier, which should solve the peak, as in the classical case. The problem arises here when we unify two variables with incomparable sorts having a common subsort. This yields a new variable in the substitution image, as the following example illustrates :

Example 12.1 *Let* $\phi_1 : f(x^{:\{A\}})^{:\emptyset} \to g(x^{:\{A\}})^{:\emptyset}$ *and* $\phi_2 : f(y^{:\{B\}})^{:\emptyset} \to y^{:\{B\}}$ *be two decorated rewrite rules in* R. *Then* $\mathcal{U} = (f(x^{:\{A\}})^{:\emptyset} \cong_d^? f(y^{:\{B\}})^{:\emptyset})$ *has the following principal solution* $\sigma = \{x^{:\{A\}} \mapsto z^{:\{\langle A,B\rangle\}}, y^{:\{B\}} \mapsto z^{:\{\langle A,B\rangle\}}\}$, *assuming* $A \bowtie_{\mathcal{S}}^{syn} B$, $\exists C \in \mathcal{S} : C \leq_{\mathcal{S}}^{syn} A, B$ *in the current presentation* $\mathcal{P}$.

Therefore the equality

$$\phi^* : g(z^{:\{\langle A,B\rangle\}})^{:\emptyset} = z^{:\{\langle A,B\rangle\}}$$

should allow the peak $g(t^{:\{A,B\}})^{:\emptyset} {}^*\!\!\leftarrow_{\phi_1} f(t^{:\{A,B\}})^{:\emptyset} \rightarrowtail_{\phi_2} t^{:\{A,B\}}$ *to converge, i.e. it should make the left and the right terms equal. But without inherited sorts, neither* $\{z^{:\{\langle A,B\rangle\}} \mapsto t^{:\{A,B\}}\}$ *nor anything equivalent can be expressed.*

Furthermore, adding inherited sorts also implies that decorated unification with $\mathbf{UNIF_d}$ becomes unitary. However, the sort inheritance notion is more general than syntactical regularity discussed in [SS87], [SNGM89] or [Wal89]:

1. Sort inheriting takes equalities into account and so may be called a "semantical" property. Thus, we have undecidability in general.

2. There are non-regular, but sort inheriting specifications like

$$((\mathcal{S} = \{A, B, C\}, \mathcal{F} = \{a\}), \mathcal{P} = \{z :: C, z : A, z : B, a : A, a : B\})$$

with $arity(a) = 0$, where a is a counter-example for regularity.

3. The construction of $\mathcal{S}_\Diamond$ is very close to the transformation of non-regular signatures into regular ones (see [SS87]). However, due to the fact that sort inheritance is semantical, the transformation cannot be completely performed a priori.

At the semantic level, sort inheritance and inherited sorts can be seen as a restriction of models of the current specification.

Hence, we refine the notion of algebras :

Definition 12.2 *A $\Sigma_\Diamond$-algebra is a pair $(|\mathcal{A}|, .^{\mathcal{A}})$ of a domain $|\mathcal{A}|$ and an interpretation function $.^{\mathcal{A}}$. $.^{\mathcal{A}}$ itself is composed of interpretations for each symbol in $\mathcal{S}$ and $\mathcal{F}$, such that :*

1. *$\forall A \in \mathcal{S}$, the interpretation of A, $A^{\mathcal{A}}$ is a non-empty set, s.t. $\Omega^{\mathcal{A}} = |\mathcal{A}|$*

2. *$\forall f \in \mathcal{F}$, the interpretation of f, $f^{\mathcal{A}}$ is a partial function $f^{\mathcal{A}} : |\mathcal{A}|^{arity(f)} \rightarrow |\mathcal{A}|$*

The interpretation of sorts in $\mathcal{S}_\Diamond$ follows from the one of $\mathcal{S}$ in the following way : $\forall S \in \mathcal{S}_\Diamond : S^{\mathcal{A}} = \bigcap_{A \in S} A^{\mathcal{A}}$.

The definition of models does not change. Remark that non-emptiness of sorts in $\mathcal{S}$ implies also the non-emptiness of sorts in $\mathcal{S}_\Diamond$, since for every sort $S \in \mathcal{S}_\Diamond$, we know that there is a $A \in \mathcal{S}$, s.t. $\langle A \rangle \leq^{syn}_{\mathcal{S}_\Diamond} S$.

A *$\Sigma_\Diamond$-homomorphism* is a G-algebra Σ-homomorphism $h : \mathcal{A} \rightarrow \mathcal{B}$ satisfying $\forall S \in \mathcal{S}_\Diamond : h(S^{\mathcal{A}}) = \bigcap_{A \in S} h(A^{\mathcal{A}})$.

It can easily be shown that there exists a free construction for all G-algebras to $\Sigma_\Diamond$-algebras with inherited sorts and therefore initial as well as free models are preserved. Adding these inherited sorts is rather easy. Proving the emptiness of all other possible sort intersections is the duty of the **Detect** rule in our completion process. Since we deal with semantical sorts, we cannot expect a decidable test.

12.2 Retracts are Superfluous

As a consequence of working with semantical sorts, the retracts defined in [GD92, JKKM92] in order to handle syntactically ill-formed terms are superfluous. Clearly, our results guarantee that the needed sort will appear eventually in the decoration of the syntactically ill-formed term, if and only if it is semantically well-formed. The following example issued from [GD92] illustrates this:

Example 12.3 *Let $\mathcal{P} = \{x :: Nat,\ y :: NeStack,\ z :: Stack,\ y : Stack,\ empty(z) : Stack,\ push(x, z) : NeStack,\ top(y) : Nat,\ pop(y) : Stack,\ top(push(x, z)) = x,\ pop(push(x, z)) = z\}$ which will be translated into:*

$$\begin{array}{rcll}
y^{:s} & \rightarrow & y^{:s \cup \{Stack\}} & \text{if } \{Stack\} \not\subseteq s \\
empty(z^{:\{Stack\}})^{:s} & \rightarrow & empty(z^{:\{Stack\}})^{:s \cup \{Stack\}} & \text{if } \{Stack\} \not\subseteq s \\
push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:s} & \rightarrow & push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:s \cup \{NeStack\}} & \text{if } \{NeStack\} \not\subseteq s \\
top(y^{:\{NeStack\}})^{:s} & \rightarrow & top(y^{:\{NeStack\}})^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
pop(y^{:\{NeStack\}})^{:s} & \rightarrow & pop(y^{:\{NeStack\}})^{:s \cup \{Stack\}} & \text{if } \{Stack\} \not\subseteq s
\end{array}$$

$$\begin{aligned} top(push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:\emptyset})^{:\emptyset} &= x^{:\{Nat\}} \\ pop(push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:\emptyset})^{:\emptyset} &= z^{:\{Stack\}} \end{aligned}$$

Composing the decoration rules, simplifying the equalities with decoration rules and finally orienting them yields:

$$\begin{aligned} push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:s} &\rightarrow push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:s \cup \{NeStack, Stack\}} \\ &\quad \text{if } \{NeStack, Stack\} \not\subseteq s \\ top(push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:\{NeStack\}})^{:\{Nat\}} &\rightarrow x^{:\{Nat\}} \\ pop(push(x^{:\{Nat\}}, z^{:\{Stack\}})^{:\{Stack\}})^{:\{Stack\}} &\rightarrow z^{:\{Stack\}} \end{aligned}$$

The completion process stops at this point and, e.g. the term

$$top(pop(push(2^{:\emptyset}, push(1^{:\emptyset}, empty^{:\emptyset})^{:\emptyset})^{:\emptyset})^{:\emptyset})^{:\emptyset}$$

cannot be typed statically on top. The normalization with decoration rules only gives $top(pop(push(2^{:\{Nat\}}, push(1^{:\{Nat\}}, empty^{:\{Stack\}})^{:\{NeStack\}})^{:\{NeStack\}})^{:\{Stack\}})^{:\emptyset}$. *But after two reduction steps we get* $1^{:\{Stack\}}$, *which has a top decoration and is therefore meaningful in any G-algebra satisfying* $\mathcal{P}$.

However, when using retracts, the term $top(pop(push(1^{:\emptyset}, empty^{:\emptyset})^{:\emptyset})^{:\emptyset})^{:\emptyset}$ *is statically typable with retracts but non-sense in the quotient algebra of* $\mathcal{P}$. *Normalizing it yields* $top(empty^{:\{Stack\}})^{:\emptyset}$, *proving that there is an algebra, in which the term does not make sense, since the top decoration is empty.*

12.3 Subsumption of Sort-Decreasing Rules Approach

We now show that our decorated completion is a conservative extension of the procedure in [GKK90]. In this framework, $\mathcal{S}$ is finite and function declarations translate to flat, linear decoration rules. Syntactic regularity of the signature is assumed and implies that every term has a least sort computed by an algorithm using the static signature. This algorithm is imitated in our approach by a bottom-up normalization process with decoration rules only.

Furthermore, the procedure in [GKK90] only orients rules if they are sort decreasing, which means that for every instance of a rule, the least sort of the right-hand side is smaller than the least sort of the left-hand side. Therefore, semantical and syntactical regularity coincide.

Proposition 12.4 *Let* $\mathcal{P}$ *be a syntactically regular, subsort unique presentation using only flat, linear term declarations, in a signature* Σ *with a finite set of sorts* $\mathcal{S}$.

Any finite completion derivation of $\mathcal{P}$, *using the order-sorted extension of the classical Knuth-Bendix algorithm ([GKK90]) with sort-decreasing rules, can be transformed into a finite decorated derivation without failure.*

Proof: The sort decreasingness hypothesis implies that there are no decoration critical pairs at all. Furthermore, simplifying eagerly with decoration rewrite rules allows for orientation with **Orient_SD** only, i.e. **Orient_NSD** gets superfluous. Consequently, there are no new decoration rules added during the whole decorated completion.

The critical pairs in $CP(D, R)$ are used to compute explicitely weakenings (restricted forms of decorated rewrite rules obtained by specialization of its variables), in order to compute the same critical pairs between rules in R. Since the number of decoration rewrite rules is finite, there can only be a finite number of such weakenings for each rule. As for simplification with decoration rules, we have to assume the computation of weakenings to be performed eagerly. Remark that a weakening can be simplified if and only if the decorated rewrite rule it stems from can be simplified, since variables are not specialized during undecorated matching.

Furthermore, no new $CP(R,R)$ critical pairs are generated by this process, since all unifiers of instantiations are covered by unifiers of the originals. In fact, the existence of a new $CP(R,R)$ critical pair would imply that we forgot one in the original deduction. This is due to the fact that any well-sorted substitution of the initial derivation implies the existence of a corresponding decorated substitution, because of the eager computation of weakenings. This eager application is also the reason why there are no additional collapses due to decoration rules.

Hence, we can imitate such a completion procedure in linear time with a factor limited by m^k, where m is the maximal number of leaves occurring in a left-hand side of an equation during the unsorted completion and k is the maximum of overloads of operators. This factor maybe minimized, if rules and specializations are treated as classes: undecorated terms in [GKK90] are in fact representatives of their class of specializations.

Finally, remark that sort-decreasingness implies both sort inheritance and the fact that $\leq_{\mathcal{S}}^{syn} = \leq_{\mathcal{S}}^{sem}$: $\mathcal{P}_\infty = (\bot, \bot, \bot)$ would be in contradiction with Theorem 11.17 and Proposition 10.10.

□

Therefore we can say that $\mathcal{OSC}$ with the results of Section 11.4 corresponds with sort-decreasing completion with flat, linear term declarations. Analogously, $\mathcal{SLC}$ in Section 11.5.1 can be interpreted as a method for completion with flat term declarations. However, both allow for temporary presence of non-flat, non-linear term declarations, handled in Section 11.6.1, and are only seen as proof method for sort inheritance. Both can and should be followed by an application of $\mathcal{OSC}$, where syntactically untypable terms in rules and equalities can temporarily exist - a problem that complicates order-sorted completion with semi-linear term declarations and syntactic sorts.

12.4 The Tree Automata Approach

Concurrently with the development of this work, H. Comon designed the completion of rewrite systems with membership constraints [Com92]. Also motivated by the failure of the critical pair lemma, his approach of the problem is to provide new deduction rules and to compute critical pairs in a fragment of second-order logic. In order to argue that our approach is not subsumed by the completion with membership constraints of [Com92], we exhibit an example that does terminate in our approach but not in the other one. Actually, this is the example already used by [CH91].

Example 12.5 *Given the rewrite system R*

$$\begin{array}{rcl} y \in S & : & f(g(y)) \to b \\ & & h(x) \to l(x) \end{array}$$

where $S = \{h^i(a)\}$, the completion procedure in [Com92] generates the infinite set of rules

$$\{X \in h^i(.) \wedge x \in S \ : \ f(g(X(l^j(x)))) \to b\}_{j=0}^{\infty}.$$

(see [CH91])

Using our decorated approach, we transform the system into

$$\begin{array}{lcl}
a^{:s} & \to & a^{:s\cup\{A\}} \text{ if } \{A\} \not\subseteq s \\
b^{:s} & \to & b^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
(x :: A)^{:s} & \to & (x :: A)^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
h(x^{:\{A\}})^{:s} & \to & h(x^{:\{A\}})^{:s\cup\{A\}} \text{ if } \{A\} \not\subseteq s \\
h(y^{:\{B\}})^{:s} & \to & h(y^{:\{B\}})^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
g(y^{:\{B\}})^{:s} & \to & g(y^{:\{B\}})^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
f(y^{:\{B\}})^{:s} & \to & f(y^{:\{B\}})^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
l(y^{:\{B\}})^{:s} & \to & l(y^{:\{B\}})^{:s\cup\{B\}} \text{ if } \{B\} \not\subseteq s \\
f(g(x^{:\{A\}})^{:\emptyset})^{:\emptyset} & = & b^{:\emptyset} \\
h(y^{:\{B\}})^{:\emptyset} & = & l(y^{:\{B\}})^{:\emptyset}
\end{array}$$

Remark that $A \leq_S^{syn} B$ and $h > l$, $f, g > b$ in the precedence of the used decorated recursive path ordering (see Definition 7.21 on how to get the decorated version) $>_d$. After some applications of **Simplify** *using decoration rules and final orientations, the last two equalities become decorated rewrite rules :*

$$\begin{array}{lcl}
f(g(x^{:\{A\}})^{:\{B\}})^{:\{B\}} & \to & b^{:\{B\}} \\
h(y^{:\{B\}})^{:\{B\}} & \to & l(y^{:\{B\}})^{:\{B\}}
\end{array}$$

The computation of $CP(D, R)$ yields now the new decorated equation $h(x^{:\{A\}})^{:\{A,B\}} = l(x^{:\{A\}})^{:\{B\}}$ that can be oriented as before giving an additional decoration rule:

$$\begin{array}{lcl}
h(x^{:\{A\}})^{:\{A,B\}} & \to & l(x^{:\{A\}})^{:\{A,B\}} \\
l(x^{:\{A\}})^{:s} & \to & l(x^{:\{A\}})^{:s\cup\{A\}} \text{ if } \{A\} \not\subseteq s
\end{array}$$

This results in a confluent decorated term rewriting system.

The following peak:

$$b \leftarrow f(g(h(h(a)))) \to f(g(l(l(a)))).$$

given in [CH91] is therefore confluent. This is because $f(g(l(l(a))))$ is normalized using the decoration rules into

$$f(g(l(l(a^{:\{A\}})^{:\{B\}})^{:\{B\}})^{:\{B\}})^{:\{B\}}$$

that can be rewritten into

$$f(g(l(l(a^{:\{A\}})^{:\{A,B\}})^{:\{A,B\}})^{:\{B\}})^{:\{B\}}$$

using $l(x^{:\{A\}})^{:s} \to l(x^{:\{A\}})^{:s\cup\{A,B\}}$ if $\{A, B\} \not\subseteq s$ twice and finally into $b^{:\{B\}}$ using $f(g(x^{:\{A\}})^{:\{B\}})^{:\{B\}} \to b^{:\{B\}}$ which is now applicable.

Nevertheless, a strictly more powerful language – due to the second order monadic logic fragment – is used in [Com92]. This power is needed for the specification of equality schemata, which is indeed not possible in our first-order approach. Let us consider the map function as an example:

Example 12.6 *Let $\perp$ be an independent constant, $\mathcal{F}_1$ be the set of unary functions in the signature and $\overline{\mathcal{F}_1}$ be the corresponding disjunction of contexts of the form $f[.]_1$ for all $f \in \mathcal{F}_1$, then using H. Comon's syntax yields:*

$$\begin{array}{ll}
X \in \overline{\mathcal{F}_1} : & map(X(\perp), cons(x, l)) = cons(X(x), map(X, l)) \\
& map(X, nil) = nil
\end{array}$$

This is indeed not expressible in our language since it is first order. A more order-sorted specific example could not be found yet, but it is possible that due to the integrated term schematization facilities, H. Comon can transform signatures in a more sophisticated way than we could. However, we did not find such an example yet, as well as we could not show that starting from a specification using first-order terms only, we could transform such a derivation into one of our approach. The latter might be possible, since all schematizations are also expressible as flat, non-linear term declarations using new sorts.

Using the syntax of H. Comon, we can express our $CP(R, D)$ critical pairs as:

Deduce 2'	$\dfrac{x \in q \wedge \phi \ : \ l \to r \quad \psi \ : \ g \to d}{x \in q' \wedge \phi' \ : \ l \to r}$	if $q_{\|_p} = g \wedge \phi \wedge \psi \stackrel{*}{\to} \phi' \wedge \sigma$ and $q' = q[\sigma(d)]_p$

This can be seen as replacement for the rule **Deduce2** in [Com92]. As this rule does not introduce new second order variables, there is no more need for **Deduce3**, if we start with a first order equality set. This is the reason why we can stay in a first-order framework while treating the same kind of problems.

However, using **Deduce2'** can obviously result in non-linear (resp. non-semi-linear) regular tree expressions, i.e. we can no more decide the intersection of two sorts. But this is necessary for constraint simplification and therefore for the decidability of unification. Consequently, we can interpret our decoration critical pairs as a semi-decision procedure for the intersection of membership constraints.

In general, one can state that due to the undecidability of intersection in non-(semi-)linear membership constraints, H. Comon has to push all $CP(R, D)$-critical pairs into new rewrite rules, calculated by **Deduce2** and **Deduce3**, instead of creating new membership constraints, which corresponds with our decoration rewrite rules. Since the variable overlaps used in [Com92] seem to cause lots of rules, we also conjecture that our approach converges more frequently. Indeed, it is less the way that critical pairs are calculated than how constraints are solved, what makes this comparison so difficult.

To conclude, the approach in [Com92] is very interesting for the term schematization facilities that it incorporates. Indeed we may wonder if seeing sorts as term schemata is the best way to cope with non-sort-decreasing rules, although it clearly allows getting rid of regularity. Nevertheless, we feel that combining the decorated rewriting approach with term schematizations can still increase expressiveness and is worth being considered.

Finally, we treat an example due to H. Comon, that comes from [Com92].

Example 12.7 *Let us first give the specification:*

Sorts :	A		*translated*	$u::A$
	B			$v::B$
	C			$x::C$
	D			$y::D$
				$z::D$
Subsorts :	$A \leq D$		*translated :*	$u:D$
	$B \leq D$			$v:D$
	$C \leq D$			$x:D$
Operators :	$a :$	$\rightarrow A$	*translated*	$a \quad :A$
	$f :$	$A \rightarrow A$		$f(u) \quad :A$
	$f :$	$B \rightarrow C$		$f(v) \quad :C$
	$f :$	$C \rightarrow C$		$f(x) \quad :C$
	$f :$	$D \rightarrow D$		$f(y) \quad :D$
	$g :$	$A \rightarrow B$		$g(u) \quad :B$
	$g :$	$D \rightarrow D$		$g(y) \quad :D$
	$h :$	$D, D \rightarrow D$		$h(y,z):D$
Rules :	$f(x:C) \rightarrow a$		*translated*	$f(x) \rightarrow a$
	$g(y:D) \rightarrow h(y,y)$			$g(y) \rightarrow h(y,y)$

The decoration rules:

$u \quad :D$	*give*	$u^{:s}$	$\rightarrow u^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(1)
$v \quad :D$		$v^{:s}$	$\rightarrow v^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(2)
$x \quad :D$		$x^{:s}$	$\rightarrow x^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(3)
$a \quad :A$		$a^{:s}$	$\rightarrow a^{:s \cup \{A\}}$	if $\{A\} \not\subseteq s$	(4)
$f(u) \quad :A$		$f(u^{:\{A\}})^{:s}$	$\rightarrow f(u^{:\{A\}})^{:s \cup \{A\}}$	if $\{A\} \not\subseteq s$	(5)
$f(v) \quad :C$		$f(v^{:\{B\}})^{:s}$	$\rightarrow f(v^{:\{B\}})^{:s \cup \{C\}}$	if $\{C\} \not\subseteq s$	(6)
$f(x) \quad :C$		$f(x^{:\{C\}})^{:s}$	$\rightarrow f(x^{:\{C\}})^{:s \cup \{C\}}$	if $\{C\} \not\subseteq s$	(7)
$f(y) \quad :D$		$f(y^{:\{D\}})^{:s}$	$\rightarrow f(y^{:\{D\}})^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(8)
$g(u) \quad :B$		$g(u^{:\{A\}})^{:s}$	$\rightarrow g(u^{:\{A\}})^{:s \cup \{B\}}$	if $\{B\} \not\subseteq s$	(9)
$g(y) \quad :D$		$g(y^{:\{D\}})^{:s}$	$\rightarrow g(y^{:\{D\}})^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(10)
$h(y,z):D$		$h(y^{:\{D\}}, z^{:\{D\}})^{:s}$	$\rightarrow h(y^{:\{D\}}, z^{:\{D\}})^{:s \cup \{D\}}$	if $\{D\} \not\subseteq s$	(11)

The decorated rules:

$f(x) \rightarrow a$	*give*	$f(x^{:\{C\}})^{:\emptyset} \rightarrow a^{:\emptyset}$	(12)
$g(y) \rightarrow h(y,y)$		$g(y^{:\{D\}})^{:\emptyset} \rightarrow h(y^{:\{D\}}, y^{:\{D\}})^{:\emptyset}$	(13)

They become after some **Collapse**, **Simplify** *and finally* **Orient**-*steps:*

$f(x^{:\{C\}})^{:\{C\}} \rightarrow a^{:\{A,C\}}$		(14)
$g(y^{:\{D\}})^{:\{D\}} \rightarrow h(y^{:\{D\}}, y^{:\{D\}})^{:\{D\}}$		(15)

together with the decoration rule

$$a^{:s} \rightarrow a^{:s \cup \{A,C\}} \text{ if } \{A,C\} \not\subseteq s \quad (16)$$

Now **Detect** *can be applied and shows that a is in the intersection of A and C, although there is no common subsort. Therefore the specification was shown to be non-sort inheriting. But introducing a new sort E with a new variable $u' :: E$, the term declarations $a : E$, $u' : A$, $u' : C$ (giving $E \leq_{\mathcal{S}}^{syn} A, C$) and the (never applicable) decoration rules*

$$u'^{:s} \to u'^{:s \cup \{A\}} \text{ if } \{A\} \not\subseteq s \quad (17)$$
$$u'^{:s} \to u'^{:s \cup \{C\}} \text{ if } \{C\} \not\subseteq s \quad (18)$$

results in a new $\mathcal{S}_\Diamond$ containing additionally $\langle A, C \rangle$ as sort with another new variable u'' and the declarations $u'' : A$, $u'' : C$ and $u' : \langle A, C \rangle$.

Hence, we have to restart the critical pairs computation. Applying **Deduce** *at (5) and (14) gives the following critical pair:*

$$f(u''^{:\{\langle A,C \rangle\}})^{:\{A,C\}} = a^{:\{A,C\}} \quad (19)$$

This can be oriented into:

$$f(u''^{:\{\langle A,C \rangle\}})^{:\{A,C\}} \to a^{:\{A,C\}} \quad (20)$$

Applying once more **Deduce** *at (9) and (15) yields:*

$$g(u^{:\{A\}})^{:\{B,D\}} = h(u^{:\{A\}}, u^{:\{A\}})^{:\{D\}}$$

This equation can be oriented into:

$$g(u^{:\{A\}})^{:\{B\}} \to h(u^{:\{A\}}, u^{:\{A\}})^{:\{B\}}$$
$$h(u^{:\{A\}}, u^{:\{A\}})^{:s} \to h(u^{:\{A\}}, u^{:\{A\}})^{:s \cup \{B\}} \text{ if } \{B\} \not\subseteq s$$

This finishes the completion. Remark that the last steps were independent from the introduced sorts. Consider now the peak from [Com92]:

$$f(f(h(a, a))) \leftarrow f(f(g(a))) \to a$$

This corresponds with

$$f(f(h(a^{:\{A,C\}}, a^{:\{A,C\}})^{:\{B\}})^{:\{C\}})^{:\{C\}} \leftarrowtail f(f(g(a^{:\{A,C\}})^{:\{B\}})^{:\{C\}})^{:\{C\}} \rightarrowtail a^{:\{A,C\}}$$

Using our approach, we get a confluent rewrite system using "only" first-order unification. The peak becomes confluent, since we can apply rule 14 *in the saturated system:*

$$f(f(h(a^{:\{A,C\}}, a^{:\{A,C\}})^{:\{B\}})^{:\{C\}})^{:\{C\}} \rightarrowtail a^{:\{A,C\}}.$$

Remark that the sort $\langle A, C \rangle$ only appears in the decoration of u'', a variable of this sort. Indeed, sorts in $\mathcal{S}_\Diamond \setminus \{\langle A \rangle \mid A \in \mathcal{S}\}$ cannot be added to any decorations. They only serve for evaluating conditions during matching, unification and evaluation of the condition of decoration rewrite rules.

12.5 The Signature Extension Approach

In [CH91], H. Chen and J. Hsiang present an order-sorted rewriting approach that allows for ill-sorted terms obtained from well-sorted ones by application of rewrite rules using syntactically well-sorted substitutions. Together with a condition called *sort-convergence*, a critical pair lemma can be obtained.

Their completion procedure constructs a sort convergent specification via *sort enrichment*, i.e. adding new sorts and function symbols when needed. The following small example from [GKK90] illustrates this:

Example 12.8 *Let* $\mathcal{P} = \{a : A, b : B, c : C, y :: B, z :: C, y : A, z : A, a = b, a = c\}$.

Then orienting the equalities into $R = \{a \to b, a \to c\}$ *results in the critical pair* $b = c$, *which is irreducible and cannot be oriented. The completion algorithm in [CH91] now adds a new constant* d *together with the rules*

$$b \to d,\ c \to d,$$

s.t. the final rule set is

$$\{a \to d, b \to d, c \to d\}.$$

The obtained system allows for equational proofs on the initial signature, but the normal form a term, for instance d, *may have no meaning from a computing point of view. Of course, there exists an interpretation for* d *which validates the equalities provable in the new presentation, but the models must be extended.*

In our approach, we get the following set of initial rules, after simplification of the equations with decoration rules and final orientation:

$$\begin{array}{lcll}
a^{:s} & \to & a^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
b^{:s} & \to & b^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s \\
c^{:s} & \to & c^{:s\cup\{C\}} & \text{if } \{C\} \not\subseteq s \\
y^{:s} & \to & y^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
z^{:s} & \to & z^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
a^{:\{A\}} & \to & b^{:\{B\}} & \\
a^{:\{A\}} & \to & c^{:\{C\}} &
\end{array}$$

The obvious critical pair is $c^{:\{C\}} = b^{:\{B\}}$, *that may be oriented using a decorated recursive path ordering based on the precedence* $a > b > c$, *yielding:*

$$\begin{array}{lcll}
b^{:\{B\}} & \to & c^{:\{B,C\}} & \\
c^{:s} & \to & c^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s
\end{array}$$

We can now apply **Detect** *on the two decoration rewrite rules for* c. *Therefore, we have to add a new sort* D *to* $\mathcal{S}$ *with a variable* $u :: D$ *and term declarations* $u : B$, $u : C$, $c : D$. *Then we add also* $\langle B, C\rangle$ *to* $\mathcal{S}_\emptyset$, *with a variable* $u' : \langle B, C\rangle$ *and the declarations* $u : \langle B, C\rangle$, $u' : B$, $u' : C$. *Since there was no variable unification necessary for the whole completion up to this point, we can simply continue with the new sorts and variables, after a translation of the new term declarations into decoration rewrite rules, giving:*

$$\begin{array}{lcll}
u^{:s} & \to & u^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s \\
u^{:s} & \to & u^{:s\cup\{C\}} & \text{if } \{C\} \not\subseteq s \\
u^{:s} & \to & u^{:s\cup\{\langle B,C\rangle\}} & \text{if } \{\langle B,C\rangle\} \not\subseteq s \\
u'^{:s} & \to & u'^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s \\
u'^{:s} & \to & u'^{:s\cup\{C\}} & \text{if } \{C\} \not\subseteq s \\
c^{:s} & \to & c^{:s\cup\{D\}} & \text{if } \{D\} \not\subseteq s
\end{array}$$

Finally, the last decoration rewrite rule subsumes the two other decoration rewrite rules for c, *finishing therefore the completion without new function symbol.*

Due to this adjunction of function symbols, the approach looks more appropriate for proving the truth of equivalences than for functional computation.

The following example gives a base for the comparison on a bigger specification, that can be solved in both approaches, with the difference that we don't need any new function symbols.

Example 12.9 *Let* $\mathcal{P}$ = {
$a : A, x :: A, x_1 :: A, x_2 :: A,$
$b : B, y :: B, y : A,$
$c : C, z :: C, z : A,$
$d : D, y' :: D, y' : B, y' : C$
$e : E, z' :: E, z' : C, z' : B$
$f(x_1, x_2) : A, g(x_1, x_2) : A, h(x_1, x_2) : A,$
$g(x, z) : D, h(x, z) : C,$
$g(y, x) : B, h(y, x) : E,$
$g(y', z') : D, h(y', z') : E,$
$f(x_1, x_2) = g(x_1, x_2), f(x_1, x_2) = h(x_1, x_2)$
}

be the initial presentation. In $\mathcal{S}_0$*, we have to add a new sort* $\langle B, C\rangle$*. Furthermore, we have to add a variable* u *and the term declarations* $u : B, u : C, y' : \langle B, C\rangle, z' : \langle B, C\rangle$*.*

After some initial applications of **Simplify_D** *and* **Orient**, *the decorated presentation has the following structure:*

$$
\begin{array}{rcll}
y^{:s} & \to & y^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
z^{:s} & \to & z^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
u^{:s} & \to & u^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s \\
u^{:s} & \to & u^{:s\cup\{C\}} & \text{if } \{C\} \not\subseteq s \\
y'^{:s} & \to & y'^{:s\cup\{\langle B,C\rangle\}} & \text{if } \{\langle B,C\rangle\} \not\subseteq s \\
z'^{:s} & \to & z'^{:s\cup\{\langle B,C\rangle\}} & \text{if } \{\langle B,C\rangle\} \not\subseteq s \\
f(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s} & \to & f(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
g(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s} & \to & g(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
h(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s} & \to & h(x_1^{:\{A\}}, x_2^{:\{A\}})^{:s\cup\{A\}} & \text{if } \{A\} \not\subseteq s \\
g(x^{:\{A\}}, z^{:\{C\}})^{:s} & \to & g(x^{:\{A\}}, z^{:\{C\}})^{:s\cup\{D\}} & \text{if } \{D\} \not\subseteq s \\
h(x^{:\{A\}}, z^{:\{C\}})^{:s} & \to & h(x^{:\{A\}}, z^{:\{C\}})^{:s\cup\{C\}} & \text{if } \{C\} \not\subseteq s \\
g(y^{:\{B\}}, x^{:\{A\}})^{:s} & \to & g(y^{:\{B\}}, x^{:\{A\}})^{:s\cup\{B\}} & \text{if } \{B\} \not\subseteq s \\
h(y^{:\{B\}}, x^{:\{A\}})^{:s} & \to & h(y^{:\{B\}}, x^{:\{A\}})^{:s\cup\{E\}} & \text{if } \{E\} \not\subseteq s \\
g(y'^{:\{D\}}, z'^{:\{E\}})^{:s} & \to & g(y'^{:\{D\}}, z'^{:\{E\}})^{:s\cup\{D\}} & \text{if } \{D\} \not\subseteq s \\
h(y'^{:\{D\}}, z'^{:\{E\}})^{:s} & \to & h(y'^{:\{D\}}, z'^{:\{E\}})^{:s\cup\{E\}} & \text{if } \{E\} \not\subseteq s \\
f(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}} & \to & g(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}} & \\
f(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}} & \to & h(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}} &
\end{array}
$$

Remark that there are neither critical pairs in $CP(D, D)$*, nor* $CP(R, D)$ *(since all rules in* D *are flat) or* $CP(D, R)$ *and all rules are interreduced. The only completion rule applicable is* **Deduce_RR** *for superposing the two decorated rewrite rules, yielding after orientation and simplification with decoration rewrite rules:*

$$g(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}} \to h(x_1^{:\{A\}}, x_2^{:\{A\}})^{:\{A\}}$$

The application of **Deduce_DR** *followed by* **Simplify_D**, **Deco_Norm** *and final orientation gives:*

$$
\begin{array}{rcll}
g(x^{:\{A\}}, z^{:\{C\}})^{:\{A,C,D\}} & \to & h(x^{:\{A\}}, z^{:\{C\}})^{:\{A,C,D\}} & \\
h(x^{:\{A\}}, z^{:\{C\}})^{:s} & \to & h(x^{:\{A\}}, z^{:\{C\}})^{:s\cup\{D\}} & \text{if } \{D\} \not\subseteq s \\
g(y^{:\{B\}}, x^{:\{A\}})^{:\{B\}} & \to & h(y^{:\{B\}}, x^{:\{A\}})^{:\{B,E\}} & \\
g(y'^{:\{D\}}, z'^{:\{E\}})^{:\{D\}} & \to & h(y'^{:\{D\}}, z'^{:\{E\}})^{:\{D,E\}} & \\
h(y'^{:\{D\}}, z'^{:\{E\}})^{:s} & \to & h(y'^{:\{D\}}, z'^{:\{E\}})^{:s\cup\{D\}} & \text{if } \{D\} \not\subseteq s
\end{array}
$$

Now, **Detect** *gets applicable to the two decoration rules for $h(y', z')$, since D and E do not have a common subsort, but $h(y', z')$ is provable to be in the intersection. Hence, a new sort G has to be introduced, s.t. $G \leq_{\mathcal{S}}^{syn} D, E$ and $h(y', z') : G$, resulting also in a new sort $\langle D, E \rangle$ in $\mathcal{S}_\diamond$, with $G \leq_{\mathcal{S}}^{syn} \langle D, E \rangle \leq_{\mathcal{S}}^{syn} D, E$. However, the recalculation of all critical pairs doesn't change anything, since no variable of sort D was unified with one of sort E.*

12.6 The T-contact Method

Another approach for the case of flat linear function declarations is described by A. Werner [Wer93]. Using the concept of semantical sorts, A. Werner proves a critical pair lemma, that allows for syntactically ill-sorted terms. This leads to a decision procedure for order-sorted equalities over extended terms, i.e. not necessarily well-formed terms, if the rewriting system is weakly sort-decreasing. The latter is the extension of sort decreasingness to multiple rewrite steps. Formally, for all semantically well-formed terms t, t', $t : A$ (syntactically) and $t \rightarrowtail_R t'$ implies that there exists some t'' with $t' \overset{*}{\rightarrowtail}_R t''$ and $t'' : A$ (syntactically). Clearly, this property solves the problem of retracts. However, if weak sort-decreasingness does not hold, the decidability cannot be obtained for all true equalities in the initial model of the specification, even with the confluence of the corresponding rewriting system. This is due to the fact, that in this case the rewrite relation becomes undecidable, since it is restricted to semantically well-formed terms and semantical sorts are proven undecidable in general, even in confluent rewrite systems. This is not in contradiction with our results, as example 12.10 shows.

Furthermore, the completion procedure given in [Wer93], transforms weakly sort-decreasing rewrite systems into sort-decreasing ones. Let us discuss this completion in full detail. Before the start of the completion procedure, any specification has to transformed into a *range unique* one, where any function symbol can only be declared to belong to exactly one range sort, but may have several different domains.

Similarly to [CH91], the rewriting relation is also defined for terms derived from syntactically typable terms under the rewrite relation, which uses so-called *T-substitutions*, which test only the coarity of the top symbols of images, instead of the classical order-sorted substitutions of [GKK90, SS87, SNGM89].

The main difference is due to the range-uniqueness of the used signature: any image of a variable $x :: A$ must be either a variable of a subsort of A or a semantically well-sorted term with a top symbol with a coarity which is a subsort of A. This allows for a critical pair lemma using also *T-contacts* of two rule's left hand sides at variable positions, but not strictly under them as in [Com92], if the replacement at the variable position is not a T-substitution.

As there are usually many variable positions in rule's left hand sides, this approach may lead to much inefficiency or even divergence. The example given in the introduction of [Wer93], reminded and extended below, seems to us a better reason for term declarations than for variable overlaps.

Example 12.10 *Given $\mathcal{P} = \{o : Nat, x :: Nat, x : Int, s(x) : Nat, y :: Int, z :: Int, sq(y) : Nat, |y| : Nat, y * z : Int, sqrt(x) : Nat, opp(y) : Int, |x| = x, sq(y) = y * y, opp(y) * opp(y) = y * y\}$, where the equalities get oriented into*

$$|x| \rightarrow x, \ sq(y) \rightarrow y * y, \ opp(y) * opp(y) \rightarrow y * y$$

we can calculate a T-overlap

$$(sq(y), |y * y|),$$

which is oriented into $sq(x) \rightarrow |x * x|$. There are no more critical pairs or T-overlaps and consequently the four rewrite rules are confluent. Hence, the peaks

$$s^n(sq(y)) \leftarrow |s^n(sq(y))| \rightarrow |s^n(y * y)| \ \blacklozenge$$

can be solved, yielding $s^n(y * y)$.

However, the rule $sq(y) \to y * y$ *is not sort decreasing and the term* $sq(y)$ *is a counter example for weak sort decreasingness, i.e. there are equalities like* $sqrt(y * y) = sqrt(opp(y) * opp(y))$ *with a common, unique normal form* $sqrt(y * y)$ *for both sides, but none of the three terms is syntactically typable. Clearly, the equality holds in the initial model of* $\mathcal{P}$, *since* $sqrt(sq(y)) = sqrt(y*y)$ *and the first term is syntactically typable. Unfortunately, the existence of such a term is in general undecidable.*

Remark that $y * y : Nat$ *is a semi-linear function declaration, that is derived automatically using our approach, when the equality* $sq(y) = y*y$ *is oriented after typing the two members using* **Simplify** *with decoration rules. In fact, we can complete* $\mathcal{P}$ *in our approach and obtain therefore the decidability of the theory of* $\mathcal{P}$.

In our approach, the initial decoration rules are:

$$\begin{array}{rcll}
x^{:s} & \to & x^{:s \cup \{Int\}} & \text{if } \{Int\} \not\subseteq s \\
o^{:s} & \to & o^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
s(x^{:\{Nat\}})^{:s'} & \to & s(x^{:\{Nat\}})^{:s' \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s' \\
opp(y^{:\{Int\}})^{:s'} & \to & opp(y^{:\{Int\}})^{:s' \cup \{Int\}} & \text{if } \{Nat\} \not\subseteq s' \\
sq(y^{:\{Int\}})^{:s} & \to & sq(y^{:\{Int\}})^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
sqrt(x^{:\{Nat\}})^{:s} & \to & sqrt(x^{:\{Nat\}})^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
|y^{:\{Int\}}|^{:s} & \to & |y^{:\{Int\}}|^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
(y^{:\{Int\}} * z^{:\{Int\}})^{:s} & \to & (y^{:\{Int\}} * z^{:\{Int\}})^{:s \cup \{Int\}} & \text{if } \{Int\} \not\subseteq s
\end{array}$$

The initial decorated equalities:

$$\begin{array}{rcl}
|x^{:\{Nat\}}|^{:\emptyset} & = & x^{:\{Nat\}} \\
sq(y^{:\{Int\}})^{:\emptyset} & = & (y^{:\{Int\}} * y^{:\{Int\}})^{:\emptyset} \\
(opp(y^{:\{Int\}})^{:\{Int\}} * opp(y^{:\{Int\}})^{:\{Int\}})^{:\{Int\}} & = & (y^{:\{Int\}} * y^{:\{Int\}})^{:\{Int\}}
\end{array}$$

After **Simplify_D**, *we get:*

$$\begin{array}{rcl}
|x^{:\{Nat\}}|^{:\{Nat\}} & = & x^{:\{Nat\}} \\
sq(y^{:\{Int\}})^{:\{Nat\}} & = & (y^{:\{Int\}} * y^{:\{Int\}})^{:\{Int\}}
\end{array}$$

Now, decorating and orienting the equalities yields:

$$\begin{array}{rcll}
|x^{:\{Nat\}}|^{:\{Nat\}} & \to & x^{:\{Nat\}} & \\
sq(y^{:\{Int\}})^{:\{Nat\}} & \to & (y^{:\{Int\}} * y^{:\{Int\}})^{:\{Nat,Int\}} & \\
(y^{:\{Int\}} * y^{:\{Int\}})^{:s} & \to & (y^{:\{Int\}} * y^{:\{Int\}})^{:s \cup \{Nat\}} & \text{if } \{Nat\} \not\subseteq s \\
(opp(y^{:\{Int\}})^{:\{Int\}} * opp(y^{:\{Int\}})^{:\{Int\}})^{:\{Int\}} & \to & (y^{:\{Int\}} * y^{:\{Int\}})^{:\{Nat,Int\}} &
\end{array}$$

This is already the final presentation, since no more completion rule is applicable. Remark that top superposition decoration critical pairs are always solved.

The equality $sqrt(y * y) = sqrt(opp(y) * opp(y))$ *can be proven, since* $sqrt(sq(y))^{:\downarrow\emptyset} \downarrow_{D \cup R=_d} sqrt(sq(opp(y)))^{:\downarrow\emptyset} \downarrow_{D \cup R=_d} sqrt((y^{:\{Int\}} * y^{:\{Int\}})^{:\{Nat,Int\}})^{:\{Nat\}}$. *The set of all true equalities in the initial model of* $\mathcal{P}$ *is decidable by Theorem 11.29.*

12.7 Other Semantic Sort Approaches

An approach really similar to ours is developed in [Wit92]. L. With gives a lemma of decidability of order-sorted unification with term declarations, based on the assumption that all Σ-*critical overlaps between term declarations* are solved, i.e. covered by already existing term declarations.

The used unification algorithm of [SS87] is only complete for regular signatures, which the author obtains via a signature transformation based on a minimal complete set of unifiers. But this

kind of sets is undecidable in general (in the presence of term declarations). Hence the signature transformation is not computable and testing if all Σ-critical overlaps between term declarations of an arbitrary signature Σ are solved becomes undecidable. However, if the signature is known to be regular, an extended version of the unification algorithm of [SS87], working with a marking process that guarantees termination, is sufficient to decide if all Σ-critical overlaps between term declarations are solved.

Remark that our approach does not give a computable transformation, too, because the completion used to check sort inheritance might not terminate. Nevertheless, if the procedure terminates, maybe after some conservative extensions of the signature (as described in Section 10.5), the signature is known to be semantically regular and the unification is therefore complete.

A last considered approach is worked out by P. Watson and J. Dick [WD89]. In order to approximate semantical sorts during completion, P. Watson and J. Dick rely on instances of equalities already generated by the completion procedure to propagate sorts. The data structures for the realization of approximated least semantical sorts, the unification algorithm and the handling of all corresponding undecidability problems are left open but we feel that they can be solved as in our approach, since the propagation of sort information can be compared to our $CP(R, D)$-critical pair computation. Furthermore, P. Watson and J. Dick propose adding intersection sorts if equal terms happen to belong to incomparable sorts, but do not give a practical algorithm for doing this.

13 Conclusion

The first contribution of this work is to give an operational semantics for G-algebra and equational deduction in an order-sorted framework. The need for retracts and sort-decreasingness disappears in our approach, which is more powerful than previous approaches such as [GKK90, Wal92], in the sense that every completion process that terminates in these frameworks also terminates with ours.

The second interest of our approach is to formalize the notion of decorations. Decorated terms appear to be quite adequate to deal with membership declarations coming either from variable declarations, or term declarations. In fact decorations are exactly what is needed at run time for recording sort updates. The operations of matching and unification proposed in this paper are limited as much as possible to a local use of this decoration information. As a consequence, this gives a theoretical model for the implementation of dynamic types in a quite efficient way via for example the use of jungle [HP88] rewriting to implement decorated terms.

The third contribution is to provide a relation with the notion of deduction with constraints [KKR90, NR92]. In the same vein, H. Comon designed the completion of rewrite systems with membership constraints [Com92]. We feel that the notion of decorated terms should provide another attractive alternative, while keeping the interesting notion of sorts as constraints.

A promising direction for further research is to extend the computations on decorated terms to a more powerful language on decorations, as in [MSS90], where operations on sorts can be specified.

Acknowledgements: We sincerely thank Uwe Waldmann and Andrea Werner for their comments on earlier versions of this work.

A Proof Transformations

For the definition of the decorated terms t, t', t'', t''' and rules ϕ, ϕ', ϕ'', see Proposition 10.3 and section 11.6, respectively.

A.1 Completeness of Completion

Orient_(N)SD :	$(t \overset{\sigma,p:S=q:S'}{\longleftrightarrow\!\!\!\rightarrow_E} t' \implies (t \overset{\sigma,p:S\to q:S\cup S'}{\rightarrowtail_{R'}} t'' \overset{0,1\sigma,\phi}{\hookleftarrow_{D'}} t')$
Deduce :	$(t' \overset{\sigma,\phi}{\hookleftarrow_{R\cup D}} t \overset{\sigma',\phi'}{\rightarrowtail_R} t'') \implies (t' \overset{\nu,\tau,p:S=q:S'}{\longleftrightarrow\!\!\!\rightarrow_{E'}} t'')$
Deduce_deco :	$(t' \overset{\sigma',\phi'}{\hookleftarrow_{D\cup R}} t \overset{\sigma,\phi}{\rightarrowtail_D} t'') \implies (t' \overset{\tau,(p:s\to p:s\cup S \text{ if } S\not\subseteq s)}{\rightarrowtail_{D'}} t'_0 \overset{0,1\sigma',\phi'}{\hookleftarrow_{D'\cup R'}} t'')$
Simplify_R :	$(t \overset{\sigma',p:S=q:S'}{\longleftrightarrow\!\!\!\rightarrow_E} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{R'}} t'' \overset{\sigma',p'':S''=q:S'}{\longleftrightarrow\!\!\!\rightarrow_{E'}} t')$
Simplify_D :	$(t \overset{\sigma',p:S=q:S'}{\longleftrightarrow\!\!\!\rightarrow_E} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{D'}} t'' \overset{\tau,p'':S''=q:S'}{\longleftrightarrow\!\!\!\rightarrow_{E'}} t')$
Delete :	$(t \overset{p:S=q:S'}{\longleftrightarrow\!\!\!\rightarrow_E} t) \implies \Phi$
Compose_R :	$(t \overset{\sigma',l:S_l\to r:S_r}{\rightarrowtail_R} t') \implies (t \overset{\sigma',l:S_l\to r':S_{r'}}{\rightarrowtail_{R'}} t'' \overset{\sigma'\circ\sigma,\phi}{\hookleftarrow_{R'}} t')$
Compose_D :	$(t \overset{\sigma',l:S_l\to r:S_r}{\rightarrowtail_R} t') \implies (t \overset{\tau,l:S_l\to r':S_{r'}}{\rightarrowtail_{R'}} t'' \overset{\sigma'\circ\sigma,\phi}{\hookleftarrow_{D'}} t')$
Compose_D_deco :	$(t \overset{\sigma',\phi'}{\rightarrowtail_D} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{D'}} t'' \overset{\tau,\phi''}{\rightarrowtail_{D'}} t''' \overset{\sigma'\circ\sigma,\phi}{\hookleftarrow_{D'}} t')$
$(\omega \neq \Lambda,\ \omega = \Lambda)$	$(t \overset{\sigma',\phi'}{\rightarrowtail_D} t') \implies (t \overset{\sigma',\phi''}{\rightarrowtail_{D'}} t'' \overset{\tau,\phi}{\hookleftarrow_{D'}} t')$
Compose_R_deco :	$(t \overset{\sigma',\phi'}{\rightarrowtail_D} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{R'}} t'' \overset{\sigma',\phi''}{\rightarrowtail_{D'}} t''' \overset{\sigma'\circ\sigma,\phi}{\hookleftarrow_{R'}} t')$
Subsume_deco :	$(t \overset{\sigma',\phi'}{\rightarrowtail_D} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{D'}} t'' \overset{0,1\tau,\phi}{\hookleftarrow_{D'}} t')$
Collapse_R :	$(t \overset{\sigma',l:S_l\to r:S_r}{\rightarrowtail_R} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{R'}} t'' \overset{\sigma',l':S_{l'}=r:S_r}{\longleftrightarrow\!\!\!\rightarrow_{E'}} t')$
Collapse_D :	$(t \overset{\sigma',l:S_l\to r:S_r}{\rightarrowtail_R} t') \implies (t \overset{\sigma'\circ\sigma,\phi}{\rightarrowtail_{D'}} t'' \overset{\tau,l':S_{l'}=r:S_r}{\longleftrightarrow\!\!\!\rightarrow_{E'}} t')$

A.2 Maximally Subterm Sharing Rewriting

Orient_(N)SD : $(t \leftarrow\!\diamond\!\rightarrow^{\sigma,p:S=q:S',O}_{E} t') \Rrightarrow (t \rightarrowtail^{\sigma',p:S\to q:S\cup S',O}_{R'} t'' \overset{0,1}{\leftarrowtail}{}^{\sigma,\phi,O}_{D'} t')$

Delete : $(t \leftarrow\!\diamond\!\rightarrow^{p:S=q:S'}_{E} t) \Rrightarrow \Phi$

Deduce_MSS : $(t' \leftarrowtail^{\sigma',\phi',O'}_{R\cup D} t \rightarrowtail^{\sigma'',\phi'',O''}_{R} t'') \Rrightarrow (t' \overset{0,1}{\rightarrowtail}{}^{\sigma,\phi,\overline{O'}}_{R'} \overline{t'} \leftarrow\!\diamond\!\rightarrow^{\tau,p:S=\phi:S'}_{E'} t''' \leftarrowtail^{\sigma,\phi}_{R'\cup D'} \overline{t''} \overset{0,1}{\leftarrowtail}{}^{\sigma'',\phi'',\overline{O''}}_{R'} t'')$

Simplify_MSS_D : $(t \leftarrow\!\diamond\!\rightarrow^{\sigma',p:S=q:S',O'}_{E} t') \Rrightarrow (t \rightarrowtail^{\sigma'\circ\sigma,\phi,O'.O}_{D'} t'' \leftarrow\!\diamond\!\rightarrow^{\sigma'\circ\sigma,p'':S''=q:S',O'}_{E'} t')$

Compose_MSS_D : $(t \rightarrowtail^{\sigma',l:S_l\to r:S_r,O'}_{R} t') \Rrightarrow (t \rightarrowtail^{\sigma'\circ\sigma,l:S_l\to r':S_{r'},O'}_{R'} t'' \leftarrowtail^{\sigma'\circ\sigma,\phi,O'.O}_{D'} t')$

Collapse_MSS_D : $(t \rightarrowtail^{\sigma',l:S_l\to r:S_r,O'}_{R} t') \Rrightarrow (t \rightarrowtail^{\sigma'\circ\sigma,\phi,O'.O}_{D'} t'' \leftarrow\!\diamond\!\rightarrow^{\sigma'\circ\sigma,l'':S_{l'}=r:S_r,O'}_{E'} t')$

Deduce_deco_MSS : $(t' \leftarrowtail^{\sigma',\phi',O'\cup O''.O}_{D\cup R} t \rightarrowtail^{\sigma,\phi,O''}_{D} t'') \Rrightarrow (t' \rightarrowtail^{\tau,(p:s\to p:s\cup S \text{ if } S\not\subseteq s),O''}_{D'} t'_0 \leftarrowtail^{\sigma',\phi',O'}_{D'\cup R'} t'')$

Subsume_deco_MSS : $(t \rightarrowtail^{\sigma',\phi',O}_{D} t') \Rrightarrow (t \rightarrowtail^{\sigma'\circ\sigma,\phi,O}_{D'} t')$

Peak w/o ovlp : $(t'{:}S' \leftarrowtail^{\phi''}_{D\cup R} t{:}S \rightarrowtail^{\phi'}_{D\cup R} t''{:}S'') \Rrightarrow (t'{:}S' \overset{0,1}{\rightarrowtail}{}^{\phi'}_{R} \overline{t'{:}S'} \rightarrowtail^{\phi''}_{D\cup R} t_0{:}S_0 \leftarrowtail^{\phi'}_{D\cup R} \overline{t''{:}S''} \overset{0,1}{\leftarrowtail}{}^{\phi''}_{R} t''{:}S'')$

Peak w/ var ovlp : $(t'{:}S' \leftarrowtail^{\phi'}_{R\cup D} t{:}S \rightarrowtail^{\phi''}_{R\cup D} t''{:}S'') \Rrightarrow (t'{:}S' \overset{0,1}{\rightarrowtail}{}^{\phi'}_{R} \overline{t'{:}S'} \rightarrowtail^{\phi''}_{D\cup R} t_0{:}S_0 \leftarrowtail^{\phi'}_{D\cup R} \overline{t''{:}S''} \overset{0,1}{\leftarrowtail}{}^{\phi''}_{R} t''{:}S'')$

References

[Bac91] L. Bachmair. *Canonical equational proofs.* Computer Science Logic, Progress in Theoretical Computer Science. Birkhäuser Verlag AG, 1991.

[BD86] L. Bachmair and N. Dershowitz. Commutation, transformation and termination. In J. Siekmann, editor, *Proceedings 8th International Conference on Automated Deduction, Oxford (UK)*, volume 230 of *Lecture Notes in Computer Science*, pages 5–20. Springer-Verlag, 1986.

[BLK+90] M. Bidoit, P. Lescanne, H. J. Kreowsky, F. Orejas, and D. Sanella, editors. *Algebraic System Specification and Development: A Survey and Annotated Bibliography*, volume 501 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

[CG91] Pierre-Louis Curien and Giorgio Ghelli. Subtyping + extensionality: Confluence of $\beta\eta$top reduction in $F_{\leq}$. In T. Ito and A. R. Meyer, editors, *Theoretical Aspects of Computer Software*, volume 526 of *Lecture Notes in Computer Science*, pages 731–749. Springer-Verlag, September 1991.

[CH91] H. Chen and J. Hsiang. Order-sorted equational specification and completion. Technical report, State University of New York at Stony Brook, November 1991.

[Com92] H. Comon. Completion of rewrite systems with membership constraints. In W. Kuich, editor, *Proceedings of ICALP 92*, volume 623 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[Dau89] M. Dauchet. Simulation of Turing machines by a left-linear rewrite rule. In N. Dershowitz, editor, *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 355 of *Lecture Notes in Computer Science*, pages 109–120. Springer-Verlag, April 1989.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In J. van Leuven, editor, *Handbook of Theoretical Computer Science*, chapter 6, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1. Equations and initial semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

[Gan91] H. Ganzinger. Order-sorted completion: the many-sorted way. *Theoretical Computer Science*, 89(1):3–32, 1991.

[GD92] J.A. Goguen and R. Diaconescu. A short survey of order-sorted algebra. *ETACS Bulletin*, 49:121–133, 1992.

[GJM85] J. A. Goguen, J.-P. Jouannaud, and J. Meseguer. Operational semantics for order-sorted algebra. In W. Brauer, editor, *Proceeding of the 12th International Colloquium on Automata, Languages and Programming, Nafplion (Greece)*, volume 194 of *Lecture Notes in Computer Science*, pages 221–231. Springer-Verlag, 1985.

[GKK90] I. Gnaedig, Claude Kirchner, and Hélène Kirchner. Equational completion in order-sorted algebras. *Theoretical Computer Science*, 72:169–202, 1990.

[GM88] J. A. Goguen and J. Meseguer. Order-sorted algebra I: Partial and overloaded operations, errors and inheritance. Technical report, SRI International, Computer Science Lab, 1988. Given as lecture at a Seminar on Types, Carnegie-Mellon University, June 1983.

[GM92] Joseph A. Goguen and José Meseguer. Order-sorted algebra I: equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 1(105):217–273, 1992.

[Gog78] J. A. Goguen. Order sorted algebras: Exceptions and error sorts, coercions and overloaded operators. Semantics and Theory of Computation Report 14, Computer Science Dept, UCLA, December 1978.

[Hin92] C. Hintermeier. Matchings and unifications in G-algebras. Rapport de DEA, Université de Nancy I, 1992.

[HK92] C. Hintermeier and Hélène Kirchner. Strict matchings and unifications in G-algebras, 1992. Centre de Recherche en Informatique de Nancy.

[HKK93] Claus Hintermeier, Claude Kirchner, and Hélène Kirchner. Operational semantics for dynamically-typed, equational specifications. Research report, CRIN, 1993.

[HP88] B. Hoffmann and D. Plump. Jungle evaluation for efficient term rewriting. In J. Grabowski, P. Lescanne, and W. Wechler, editors, *Proceedings 1st International Workshop on Algebraic and Logic Programming*, Berlin, DDR, 1988. Akademie-Verlag. To be published by Springer-Verlag.

[Hue76] G. Huet. *Résolution d'equations dans les langages d'ordre 1,2, ...,ω*. Thèse de Doctorat d'Etat, Université de Paris 7 (France), 1976.

[Hue80] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the Association for Computing Machinery*, 27(4):797–821, October 1980. Preliminary version in 18th Symposium on Foundations of Computer Science, IEEE, 1977.

[JK86] J.-P. Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15(4):1155–1194, 1986. Preliminary version in Proceedings 11th ACM Symposium on Principles of Programming Languages, Salt Lake City (USA), 1984.

[JK91] J.-P. Jouannaud and Claude Kirchner. Solving equations in abstract algebras: a rule-based survey of unification. In Jean-Louis Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 8, pages 257–321. MIT Press, Cambridge (MA, USA), 1991.

[JKKM92] J.-P. Jouannaud, Claude Kirchner, Hélène Kirchner, and A. Mégrelis. Programming with equalities, subsorts, overloading and parameterization in OBJ. *Journal of Logic Programming*, 12(3):257–280, February 1992.

[JS92] R. D. Jenks and R. S. Sutor. *Axiom: The scientific Computation System*. Springer-Verlag, Oxford, 1992.

[KB70] Donald E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[KKM88] Claude Kirchner, Hélène Kirchner, and J. Meseguer. Operational semantics of OBJ-3. In *Proceedings of 15th International Colloquium on Automata, Languages and Programming*, volume 317 of *Lecture Notes in Computer Science*, pages 287–301. Springer-Verlag, 1988.

[KKR90] Claude Kirchner, Hélène Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52, 1990. Special issue on Automatic Deduction.

[KKV93] Claude Kirchner, Hélène Kirchner, and M. Vittek. Implementing computational systems with constraints. In Paris Kanellakis, Jean-Louis Lassez, and Vijay Saraswat, editors, *Proceedings of the first Workshop on Principles and Practice of Constraint Programming, Providence (R.I., USA)*, pages 166–175. Brown University, 1993.

[Még90] Aristide Mégrelis. *Algèbre galactique — Un procédé de calcul formel, relatif aux semi-fonctions, à l'inclusion et à l'égalité.* Thèse de Doctorat d'Université, Université de Nancy I, 1990.

[Mes92] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[Mio93] Alfonso Miola, editor. *Design and Implementation of Symbolic Computation Systems.* Springer-Verlag, September 1993.

[Mos89] Peter D. Mosses. Unified algebras and institutions. In *Proceedings 4th IEEE Symposium on Logic in Computer Science, Pacific Grove*, pages 304–312, 1989.

[MSS90] V. Manca, A. Salibra, and G. Scollo. Equational type logic. *Theoretical Computer Science*, 77(1-2):131–159, 1990.

[NR92] R. Nieuwenhuis and A. Rubio. Basic superposition is complete. In B. Krieg-Brückner, editor, *Proceedings of ESOP'92*, volume 582 of *Lecture Notes in Computer Science*, pages 371–389. Springer-Verlag, 1992.

[Obe62] A. Oberschelp. Untersuchungen zur mehrsortigen quantorenlogik. *Math. Annalen*, 145(1):297–333, 1962.

[Smo89] G. Smolka. *Logic Programming over Polymorphically Order-Sorted Types.* PhD thesis, FB Informatik, Universität Kaiserslautern, Germany, 1989.

[SNGM89] G. Smolka, W. Nutt, J. A. Goguen, and J. Meseguer. Order-sorted equational computation. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Volume 2: Rewriting Techniques*, pages 297–367. Academic Press, 1989.

[SS87] M. Schmidt-Schauß. *Computational Aspects of an Order-Sorted Logic with Term Declarations.* PhD thesis, Universität Kaiserslautern (Germany), 1987.

[Uri92] T.E. Uribe. Sorted unification using set constraints. In D. Kapur, editor, *Proceedings 11th International Conference on Automated Deduction, Saratoga Springs (N.Y., USA)*, volume 607 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, June 1992.

[Wal89] U. Waldmann. Unification in order-sorted signatures. Technical Report 298, Universität Dortmund (Germany), 1989.

[Wal92] U. Waldmann. Semantics in order-sorted specifications. *Theoretical Computer Science*, 94(1):1–33, 1992.

[WD89] P. Watson and J. Dick. Least sorts in order-sorted term rewriting. Technical report, Royal Holloway and Bedford New College, University of London, 1989.

[Wer93] A. Werner. A semantic approach to order-sorted rewriting. In C. Kirchner, editor, *Proceedings 5th Conference on Rewriting Techniques and Applications, Montreal (Canada)*, volume 690 of *Lecture Notes in Computer Science*, pages 47–61. Springer-Verlag, 1993.

[Wit92] L. With. Completeness and confluence of order-sorted term rewriting. In M. Rusinowitch and J.-L. Rémy, editors, *Proceedings 3rd International Workshop on Conditional Rewriting Systems, Pont-à-Mousson (France)*, Lecture Notes in Computer Science. Springer-Verlag, July 1992.

Index

Unité de Recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)

Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399